1996-03

Implementation and evaluation of commercial off-the-shelf (COTS) voice recognition software as an input device in a windows-type environmen

West, Timothy J.

Monterey, California. Naval Postgraduate School

NAVAL POSTGRADUATE SCHOOL
MONTEREY, CALIFORNIA

THESIS

IMPLEMENTATION AND EVALUATION OF COMMERCIAL OFF-THE-SHELF (COTS) VOICE RECOGNITION SOFTWARE AS AN INPUT DEVICE IN A WINDOWS-TYPE ENVIRONMENT

by

Timothy J. West

March, 1996

Thesis Advisor:	Monique P. Fargues
Associate Advisor:	James C. Emery

Approved for public release; distribution is unlimited.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503

1. AGENCY USE ONLY *(Leave Blank)*	2. REPORT DATE March 1996	3. REPORT TYPE AND DATES COVERED Master's Thesis

4. TITLE AND SUBTITLE

IMPLEMENTATION AND EVALUATION OF COMMERCIAL OFF-THE-SHELF (COTS) VOICE RECOGNITION SOFTWARE AS AN INPUT DEVICE IN A WINDOWS-TYPE ENVIRONMENT

5. FUNDING NUMBERS

6. AUTHOR(S)

West, Timothy J.

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Naval Postgraduate School
Monterey, CA 93943-5000

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSORING / MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

12a. DISTRIBUTION / AVAILABILITY STATEMENT

Approved for public release, distribution is unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)

This thesis investigates the implementation and evaluation of commercial off-the-shelf (COTS) voice recognition as an input interface within a windows-type environment. The three software packages implemented and evaluated are DragonDictate For Windows version 1.3, VoicePilot 2.0 (both manufactured by Dragon Systems, Inc.), and IN^3 Voice Command for SPARCstation version 2.2.2 by Command Corp. VoicePilot and DragonDictate are both installed on PCs running MS Windows 3.1, and IN^3 is installed on a SPARCstation running OpenWindows 3 and SunOS 4.1.3. Several applications are manipulated using voice recognition with these three software packages. The results of this study show that DragonDictate has the most flexibility and ease of use as an input device for a windows-type environment. It is also shown that as usage increases, DragonDictates recognition accuracy is able to be improved to above 98 %. Other areas of future research are also suggested.

14. SUBJECT TERMS

Voice Recognition, Automatic Speech Recognition, DragonDictate, VoicePilot, IN^3 Voice Command, Dictation, Voice Navigation

15. NUMBER OF PAGES

84

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT UL

Approved for public release; distribution is unlimited.

IMPLEMENTATION AND EVALUATION OF COMMERCIAL OFF-THE-SHELF (COTS) VOICE RECOGNITION SOFTWARE AS AN INPUT DEVICE IN A WINDOWS-TYPE ENVIRONMENT

Timothy J. West
Lieutenant, United States Navy
B.A., Virginia Military Institute, 1988

Submitted in partial fulfillment
of the requirements for the degree of

MASTER OF SCIENCE IN INFORMATION TECHNOLOGY MANAGEMENT

from the

NAVAL POSTGRADUATE SCHOOL
March, 1996

ABSTRACT

This thesis investigates the implementation and evaluation of commercial off-the-shelf (COTS) voice recognition as an input interface within a windows-type environment. The three software packages implemented and evaluated are DragonDictate For Windows version 1.3, VoicePilot 2.0 (both manufactured by Dragon Systems, Inc.), and IN^3 Voice Command for SPARCstation version 2.2.2 by Command Corp. VoicePilot and DragonDictate are both installed on PCs running MS Windows 3.1, and IN^3 is installed on a SPARCstation running OpenWindows 3 and SunOS 4.1.3. Several applications are manipulated using voice recognition with these three software packages. The results of this study show that DragonDictate has the most flexibility and ease of use as an input device for a windows-type environment. It is also shown that as usage increases, DragonDictates recognition accuracy is able to be improved to above 98 %. Other areas of future research are also suggested.

TABLE OF CONTENTS

LIST OF FIGURES

I. INTRODUCTION

A. VOICE RECOGNITION AND C4I

In the past few years the Department of Defense (DoD) has placed an emphasis on C4I (Command, Control, Communications, Computers, and Intelligence) for military applications. An example of this is the issuance of many Service plans and directives on the implementation of C4I within each of the major services. C4I is the future for all the military services, and is playing a major role in the planning of future capabilities, makeup, and budgetary issues within DoD.

To get a better look at what is expected from C4I, let us take a look at the infrastructure of C4I. The C4I infrastructure for the Warrior is broken down into three major areas: the warrior terminal, the Warrior's battlespace, and the Infosphere (a global military and commercial communications systems and network of information databases and fusion centers accessible by the warrior from anywhere at anytime [Ref. 18: p. 10]). We will concentrate on the Warrior terminal. The Director of C4 Systems, J-6, for the Joint Staff describes the Warrior terminal as follows [Ref. 18: p. 9]:

> The Warrior's terminal is the processing equipment that will allow the Warrior to store all required on-site information and share information in multimedia forms among other terminals when required. The C4I terminal devices and their capabilities must be familiar to the Warrior. This requires the terminal to have "manprint" (look, touch, feel) that is recognizable to the user whether in the Pentagon or in the field. The terminal device may be phone size, wrist watch size or even smaller as technology develops. The terminal must satisfy the Warrior's needs of any time, any location, and any mission. The terminal will be tailored to the Warrior to best assist him or her in accomplishing the mission....

Looking closer at the Warrior terminal, we can focus on the "manprint" and multimedia. In order to give the Warrior terminal a familiar look, touch and feel of a terminal that is easily recognized by anyone wishing to operate the terminal the interface between man and machine must be natural in its implementation. The natural interface for the machine (in our case the computer) is digital in nature. The natural mode for communication for man is speech. To bridge this difference in forms of communication a device for transforming or translating speech to digital signals is required. For computers, voice recognition software and microphones are the obvious answer to this problem. By using voice recognition software, the Warrior would be able to speak to the computer and to have the computer process his or her commands. With voice recognition the Warrior will be able to navigate through the applications available on the computer and will also able to dictate letters, memos, directives, etc.

This study will show examples of commercial off-the-shelf voice recognition software, capabilities, and implementation. Each software package will be evaluated and the results given in the conclusion. The software packages evaluated will be DragonDictate and VoicePilot by Dragon Systems, Inc., and In-Cubed (IN^3) by Command Corp. These packages were selected for study because they did not require any proprietary equipment.

B. THE VOICE RECOGNITION INDUSTRY

1. The Market

Voice recognition technology has made tremendous strides in the past few years. Several major areas of commercial applications of voice recognition are dictation, personal computer interfaces, inventory maintenance, automated telephone services, and special-purpose industrial applications. The use of voice recognition in private and public telephone companies is enjoying a tremendous amount of success. Voice recognition in telecommunications is becoming a very lucrative market, averaging 40.4% annual gain in the Automatic Speech Recognition (ASR) Market. The overall market for automatic speech recognition/voice recognition (ASR/VR) technology is expected to have an annual growth of about 35% up to the year 1997 [Ref. 6: p. 57].

2. Commercial Vendors and Uses of Voice Recognition

Several vendors are producing voice recognition packages and application development products. PCvoice Inc., BBN HARK Systems Corp., Speech Systems Inc., Dragon Systems Inc., Kurzweil Applied Intelligence Inc., IBM, Microsoft, Voice Processing Corp., and Wildfire Communications Inc. have all released new voice recognition packages this year. Both Macintosh and IBM are releasing computer systems with voice recognition software included with the normal systems setup. WordPerfect Corporation has teamed with Dragon Systems to develop voice controlled word processing software and other Windows-based software. This influx of voice recognition software and applications is an indication that voice recognition is becoming more popular as an interface device as the technology improves. Already on the market are voice-activated controls for videocassette recorders, televisions, cellular phones that dial a number when the user speaks the name of a person, and multimedia games, training programs, and educational applications that respond to voice commands.

IBM Personal Dictation System has overcome a lot of the hurdles faced by all recognition software: recognition accuracy, command decoding speed, and vocabulary size. It boasts a 95 to 98% recognition accuracy, which is about one mistake out of every 20 words spoken. It is able to handle up to 90 words a minute; average speaking speed in a normal conversation is 80 words per minute. It comes with a 60,000-word vocabulary that is customizable to incorporate job-oriented jargon. The vocabulary is also expandable: with user-defined words it is able to accommodate up 82,000 words. IBM retails this product for $995. This price includes the proprietary card marketed by IBM.

For command and control systems there are many options available from the aforementioned vendors that exhibit remarkable accuracy, speed, and vocabulary size for commercial needs. The HARK Recognizer immediately comes to mind. Dr. Phillip F.

Carrigan, marketing director at UFA Inc., a developer of air traffic control simulation systems, states that [Ref. 9: p. 9]:

> The HARK Recognizer is the most mature, stable and robust speaker-independent product available... We depend on HARK products to handle the complex task of moving simulated aircraft in response to spoken commands...

Telecommunication technology is leading the way in the use of voice recognition technology. Telephone services are boasting a projected savings of hundreds of millions of dollars. AT&T and Sprint already offer voice recognition-controlled services. Sprint even offers voice activated phone cards.

C. CURRENT DOD INVOLVEMENT IN VOICE RECOGNITION

The Department of Defense has begun to incorporate voice recognition into some or its information systems. In comparison to many major civilian organizations that have incorporated voice recognition into their information system technology, the DoD is not very far behind the level of implementation in industry. Many companies have successfully integrated voice recognition into their security systems, word processing packages, and even in their telecommunications. AT&T already boasts on their television commercials that they will be bringing technology that will allow you access to your home via voice. "Smart" homes are being built that will turn on the stereo, start dinner, or even turn on any other appliance by voice command. Using computer control, one can do these things over the telephone lines from a remote location.

Currently the United States Air Force Rome, NY, Laboratory and three affiliated labs are developing systems that automatically identify individual speakers and the language being spoken [Ref. 7: p. 57]. Monitoring of enemy radio signals and enhanced analysis of aircraft accidents are two applications also being developed by the USAF. Other military applications being explored are smart cockpits, allowing the pilot to orally instruct a computer to take a selected course of action rather than flipping a switch; and in command and control, to orally instruct a computer rather than use a keyboard [Ref. 7: p. 57]. The United States Navy is currently developing an Aegis Combat Information Center system that would be operated using voice commands.

1. Advantages and Disadvantages

By automating data input and retrieval using voice recognition, the DoD would be able eliminate the need for many administrative types that do most of the data retrieval and input used by the current manual systems. Improved telecommunication service and information systems interfaces are in keeping with improving DoD information systems (IS) technology. With the migration of call control from private branch exchanges (PBX) to the computing environment as computer telephony integration (CTI) evolves, the need for voice recognition software will increase as call centers' role diminishes [Ref. 11: p. 51].

The cost for a viable voice recognition system is very small in comparison to the benefits of implementing the system. A typical commercially available system for command and control can range from less than a $100 one-time cost (for systems such as Microsoft's Sound System for Windows, Creative Lab's VoiceAssist, and Covox's Speech Blaster) to more than $10,000 annually (for systems such as BBN HARK Systems' Recognizer 2.0 Developers Toolkit and technical support from BBN HARK). With decreasing costs and increasing processor power of the newest personal computers, the costs of voice recognition software are decreasing. IBM, VERBEX, Kurzweil Applied Intelligence Inc., and other voice recognition software\hardware development industries are cutting prices for their product by as much as 50%. All of these systems support Windows (version 3.x and eventually Windows '95), OS/2, DOS, and UNIX operating systems on IBM compatible PCs, Sun workstations, Hewlett Packard and Silicon Graphics platforms. Initial investment would be minimal for implementation throughout the DoD and its service components. The only requirement would be for the acquisition and implementation of the software\hardware required for the actual voice recognition system. Most deployed and shore-based DoD assets have access to or are already on IBM or UNIX systems.

Included in the cost would be a minimum of twenty to thirty minutes lost productivity while personnel "enroll" in the discrete user-dependent and some speaker-adaptive systems. Enrolling entails a training period in which the user inputs spoken commands into the software in order to build the library of statistical models. The IBM Personal Dictation System, for example, requires the user to read a Mark Twain short story in order to "learn" the user's speech patterns. This time period would not be necessary for most continuous, speaker-independent systems, which allow the user to start giving voice commands immediately after installation.

Computer manufacturers are proceeding in their development with the assumption that speech will become an important component of the computer interface [Ref. 5: p. 54]. Near-term opportunities in voice recognition include:

1. Speech as a shortcut. Rather than opening a file by traversing many levels of hierarchy with multiple key strokes, the user just has to say "Open budget." An even timelier example is "Open the address book and call my barber." By incorporating intelligence and macros into the voice recognition software that it is possible to gain greater flexibility.

2. Hands busy/eyes busy environments are easily adaptable to voice recognition systems. An air traffic controller could give commands to his computer while steadily scanning his equipment and the skies. Inventory managers, and weapons and ammunition control officers could simply speak into a portable system instead of carrying multiple sheets of inventory and ammunition records. Roving watch standers who take readings on machinery and soundings from tanks could simply speak into a portable system and cut their roving time by a third.

3. Portability. Once a user is enrolled in a particular system, he could simply download his file and upload it into another system that utilizes the same or

compatible voice recognition software. A person could be transferred to many duty stations and never have to re-enroll on a voice recognition system.

The Naval Postgraduate School thesis by Earl Hill and Leo Kotowski further lists the advantages of voice recognition and separate them into three categories: engineering, psychological, and physiological [Ref. 10: pp. 35-38]:

A. Engineering
    1. Advantages
        a) Can be faster than other [input] modes.
        b) Can be more accurate than other [input] modes.
        c) Compatible with communications systems (telephone).
        d) Can reduce manpower requirements.
    2. Disadvantages
        a) Possible interference from noise, distortions, and competing talkers.
        b) Physical conditions (vibrations and physical orientation of speaker) may change speech patterns.
        c) No permanent record of speech (unless explicitly recorded).
        d) Microphones needed for speech input, and acoustic speakers needed for speech output.

B. Psychological
    1. Advantages
        a) Most natural form of human communication.
        b) Best for group problem solving.
        c) Universal among humans.
        d) Can reduce visual information overload.
        e) Increases in value when person is engaged in complex thought processes.
    2. Disadvantages
        a) Speech is not private; others may eavesdrop.
        b) Psychological changes (stress) may change one's speech characteristics.
        c) Speech synthesis may interfere with other aural indicators.

C. Physiological

1. Advantages
    - a) Requires less effort and motor activity than other [input] modes.
    - b) Frees the hands and eyes.
    - c) Permits multimodal operation.
    - d) Feasible in darkened area.
    - e) Is omnidirectional; does not require direct line of sight between user and ASR system.
    - f) Permits operator mobility.
    - g) Contains information on identity and emotional state of the speaker.
    - h) Contains information on the physical state of the speaker.
    - i) Simultaneous interaction with man and machines.
2. Disadvantages
    - a) Prolonged speaking may cause fatigue, which may in turn change speech characteristics.
    - b) Illness may change speech characteristics.

Studies have been performed both at the Naval Postgraduate School and by others that demonstrate and support the definite advantages of speech input over other currently available forms of input. These include reports on the effects of stress and changing environments on the user of various recognition systems (most of these were performed by the late Gary K. Poock, formerly a professor with the Systems Management department at the Naval Postgraduate School), the effect of feedback on users of ASR equipment, and the effects of various background noises on ASR systems recognition capabilities.

D. SUMMARY

Organizations using speech technology properly can enjoy enormous savings. The US Postal Service, for instance, projects that it will save $30 million by using a voice recognition system for mail sorting [Ref. 8: p. 52]. AT&T reportedly could save as much as $100 million annually by using speech recognition technology to replace up to 17,000 human operators; the company has already used the technology to eliminate 2,000 operators [Ref. 8: p. 52]. The DoD could achieve similar savings by utilizing voice recognition technology in its information systems. It would eliminate the need for most Personnelmen and Yeomen and other administrative rates since it would require fewer personnel to maintain computer-based records and to dictate letters and memos. Most Commanding Officers and Department Heads could dictate and send their own messages and letters using voice recognition technology.

Many of the disadvantages connected to voice recognition and its usage as a means of data input can be overcome by engineering and/or controlling the environment. Many of the physiological advantages work toward easing the stress and fatigue on the user enabling him to become more effective and versatile in a C4I environment. This thesis will cover the implementation and evaluation of three voice recognition software packages currently available commercially. The evaluation will cover their usage in a windows type environment within their respective required operating systems.

II. AN INTRODUCTION TO VOICE RECOGNITION

A. THE BASICS OF VOICE RECOGNITION

Voice recognition (VR), also called Automatic Speech Recognition (ASR), is the ability of speech software and hardware to convert spoken words into text or commands. Voice recognition requires the use of an analog to-to-digital (A/D) converter with the remaining computations (using a complex algorithm) taking place on a general-purpose computer. Voice recognition systems match a transform of incoming speech against a representation stored in some form of permanent memory [Ref. 5: p. 2]. A recognizer will make use of acoustic models that capture phonetic or word-level properties of speech and often a statistical model that captures the syntactic and semantic regularities of language in a particular domain [Ref. 5: p. 52]. Most leading technologies use a Hidden Markov Model (HMM) algorithm, or a Neural Network/Hidden Markov Hybrid System. The neural Network/Hidden Markov Hybrid System is used to improve inaccuracies in the HMM that are caused because [Ref. 13: 6/12/95]

> ...traditional HMMs make some false assumptions, e.g., that speech features occurring at one time are uncorrelated, and independent of other recently occurring features (even ten milliseconds earlier). SRI has developed a hybrid neural network/hidden Markov model speech recognizer that improves the accuracy of traditional HMM by modeling correlations among simultaneously occurring speech features and between current and recent features. Future work involves modeling longer-term correlations, using better basic speech features, and integrating higher-level linguistic constraints.

Voice recognition systems are categorized along a number of standard dimensions. Where a system falls in these dimensions strongly determines a system's capabilities. These dimensions are *speaker-dependent* or *speaker independent*, *dictation* or *navigation* software, *continuous* or *discrete* recognition, and small or large vocabulary. Normal human speech is continuous, with an unlimited vocabulary, and speaker independent, but in many applications none of these characteristics is required [Ref. 12: p. 35.4].

1. Speaker-Dependent (SD) Vs. Independent (SI)

A speaker-dependent system is trained to a particular voice, whereas an independent system is able to recognize the speech of many different individuals without training. Also available are speaker-adaptive systems that operate as SI systems but adapt to the speech patterns of an individual with more use, with a concomitant increase in recognition accuracy. Speaker independent systems are difficult to produce because of the differences in accent, pitch, inflection, etc. Thus most commercially available systems are speaker-dependent.

The type of training needed for significant accuracy in speaker-dependent systems requires the user to repeat each word a number of times. This is especially true for the systems with small vocabularies. The speaker-dependent system then uses this information to create a model of the word and incorporates a variability factor that accounts for slight changes in pronunciation for each utterance (Figure 1).

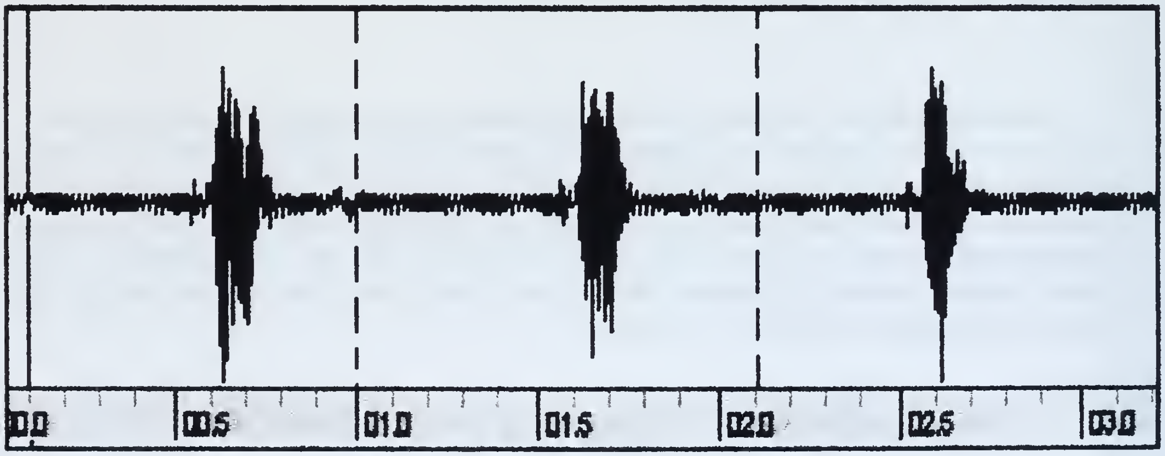


Figure 1. Three utterances of the word "*cut*," sampled at 44 kHz. (X-axis is time in seconds.)

2. Discrete Vs. Continuous Recognition

The recognition type determines if a user needs to separate individual words by short silences. Discrete Recognition or independent word recognition (IWR) systems are easier to implement because the system knows the exact extent of the word and can use this information to improve decoding accuracy. Continuous recognition is far more difficult since there are extremely small or no break at all between the utterances of words in a particular phrase. This makes it extremely difficult for the software to correctly decode the words in the phrase.

3. Vocabulary Size

The System is able to better recognize a word if the vocabulary is very small. This is because there are fewer alternative words from which the system has to choose. The vocabulary size also determines the choice of algorithm and the details of implementation [Ref. 5: p. 54]. Most small vocabulary software contains about 1000 words in their vocabulary. Larger systems handle anywhere from about 20,000 to 70,000 words.

Many of the commercially available small vocabulary systems handle several vocabularies. They do this by loading the individual vocabularies of the applications that it can control. It arranges the vocabularies in a tree-structured fashion. The words or commands that are used to start or end each application are stored in the root of the structure. When the system recognizes a word that begins an application, it retrieves the specific vocabulary for that application and makes it the active vocabulary. In a windows-type environment where there is multitasking, the vocabulary for the active window is selected.

Large vocabulary systems require a different training mechanism. It is impractical to repeat thousands of words thousands of times. Large vocabulary systems do not recognize words in the same manner as small vocabulary systems. They base their recognition schemes on elements smaller than a word such as syllables and phonemes

[Ref. 12: p. 35.5]. Because the actual pronunciation of a particular phoneme is subject to the surrounding phonemes and its corresponding allophones, it is possible to use a small number of phonemes to represent a large number of words. Only around 40 phonemes are required to speak in the English language, which is over 40,000 words [Ref. 12: p. 35.6].

O'Shaughnessy [Ref. 14] describes in detail how phonemes work. The following is a brief synopsis of the basics. The articulation of a phoneme produces a physical sound called a *phone*. An infinite number of phones can correspond to any particular phoneme because the vocal tract can vary in an infinite number of ways. *Allophones* are a class of phones corresponding to a specific variant of a phoneme. [Ref. 14: p. 56].

The ideal voice recognition system is a system that is speaker independent, supports continuous speech, has a very large vocabulary (about 60,000 or more words), and uses synthesized speech as an interface between the computer and the user. This ideal system is not yet realized in practice.

4. Dictation Vs. Navigation Software

Dictation is the process of using voice recognition as an input method when using word processing software. There are really two types of voice dictation systems that can be envisioned, differentiated by where the user's attention is focused. In the classic voice-activated-typewriter case, the user is focused on both the computer and the information being input into the system. This enables practically immediate error correction, and the system is able to prompt the user for information in the case of unclear or ambiguously identified words. The other case is when the user has his attention focused elsewhere, and he is basically "thinking out loud" and the computer is capturing those thoughts.

Navigation software, or voice command software, is used to open and close applications within the operating environment. It is also used to perform menu commands within those applications. This type of software is basically a command and control tool that is activated by voice. An example of this is the Microsoft Sound System

For Windows VoicePilot software application. This application is used within the Windows operating system to "navigate" by opening and closing windows compatible applications. It captures commands from the menus of the applications and adds them to a specific vocabulary which it creates for that particular application.

III. DRAGONDICTATE FOR WINDOWS VERSION 1.3

A. DESCRIPTION

DragonDictate is a combined navigator/dictation software package. Version 2.0 is the latest version offered (available since January 1996). The particular version implemented is the Classic Version which uses a 30,000 word vocabulary. DragonDictate version 1.3 was installed on an IBM PC compatible computer, with a Pentium processor running at 90 Mhz, 16 MB of RAM, a Sound Blaster 16 sound card, and a color monitor.

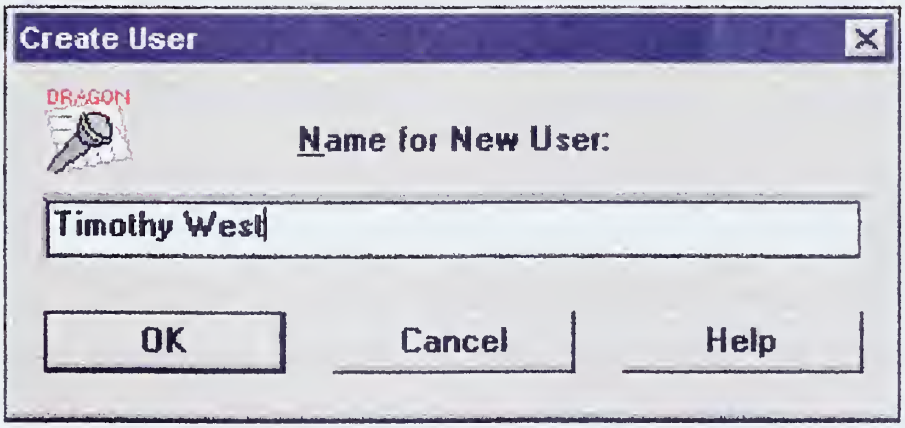
1. Installation and Setting Up

Installation of DragonDictate for Windows is very simple. The instructions included with the software are very clear and concise. Installation of the software is similar to the installation of any other Microsoft Windows application. The primary diskette is inserted into the primary drive while the user is working in Windows. While in Windows go to the Windows program manager and click on File|Run and type A:\setup.exe, where "A" is whatever your primary drive is called. The installation program will begin and all that is required is to follow the on-screen instructions. It is recommended to preload everything when given the choice, because this will enable the user to add new users without having to bother with inserting any diskettes after the initial installation is completed.

The entire program will require about 24 megabytes of hard disk space and about 12 megabytes of RAM (if you plan on having more than two users). The users guide lists the following system requirements for installing DragonDictate Classic edition [Ref. 2, pp. 2-3]:

1. One of the following sound cards:
   - A. IBM® M-ACPA (M-Audio Capture and Playback Adapter)
   - B. Creative Labs, Inc. Sound Blaster 16™
   - C. Media Vision™ ProAudio Studio 16™
   - D. Microsoft® Windows™ Sound System
2. At least in IBM 486/33mhz PC or compatible computer
3. And the following requirements for the Classic edition (30,000 words):
   - A. 24MB + 9MB per user after the first user
   - B. 10.5MB RAM which includes 3 MB of memory required by Windows
4. 3.5 inch, 1.44 MB (high-density) floppy drive
5. Microsoft Windows 3.1
6. MS-DOS or PC DOS, version 3.1 or higher
7. Color or grayscale monitor
8. Mouse recommended

After installation, DragonDictate prompts the user to enter a name for the individual that will be utilizing the software (Figure 2). This is a required step in order for the user to begin using the software. The software will create a user profile and install a minimum vocabulary for the particular user specified. DragonDictate will also prompt the

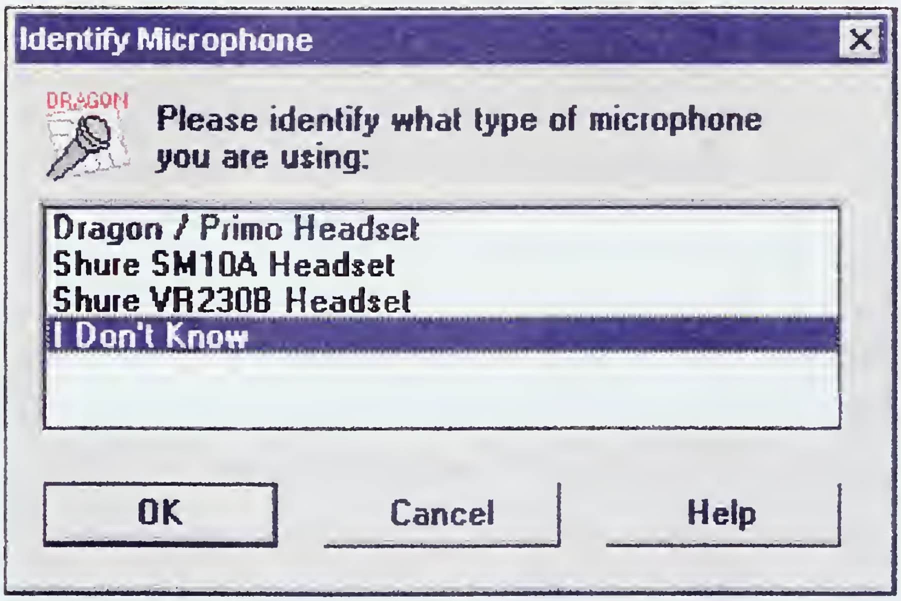


Name for New User:

Timothy West

Figure 2. Adding A New user

user to identify the type microphone/headset to be used in conjunction with the software (Figure 3). The user is allowed to select from three types of headsets which include two Shure models (SM10A headset, and the VR230B headset), one Dragon Systems headset (the Dragon/Primo headset), and a selection label "I don't know." The two Shure models are recommended because they are particularly sensitive to sound and tend to produce very good quality input for the software. The user is then asked to go through a tutorial (Fig 4.) and to perform the "Quick Training."



Identify Microphone

Please identify what type of microphone you are using:

Dragon / Primo Headset
Shure SM10A Headset
Shure VR230B Headset
I Don't Know

Figure 3. Microphone Selection

It is highly recommended to go through the tutorial. The tutorial gives the user a quick crash course in simple commands and dictation practice for use in DragonDictate. This gives the user a feel of how the software behaves and how it interacts with different applications including Windows notepad and calculator. After completing the tutorial, DragonDictate asks if you would like to do the "Quick Training." It is recommended to do the quick training session at this time. This is a required step in order for DragonDictate to recognize your speech and it also makes DragonDictate easier to use.

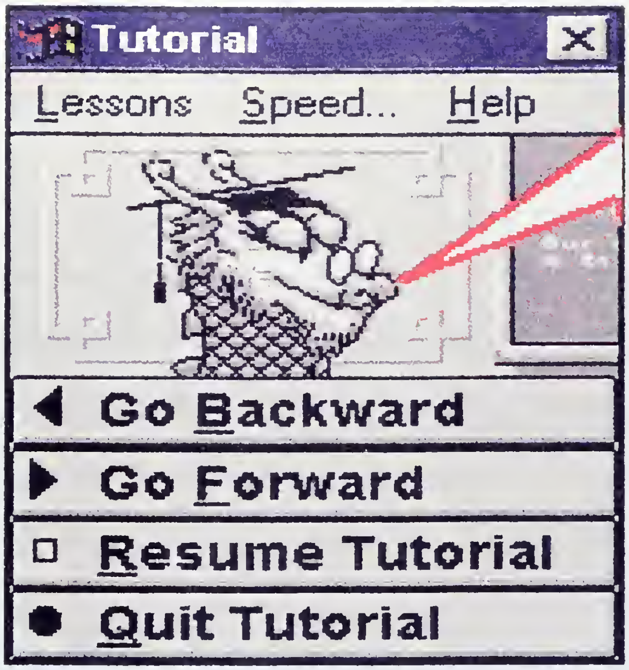


Figure 4. Tutorial Window

2. Training

Training is required once you have created a user profile. The Quick Training Window (Figure 5) allows you to set the intensity of the training. You are also to set only the repetition level and to enable or disable the "Only Listen for Word Being Trained" selection. Total training time is about 20 minutes at the default setting (Light), but may take up to 90 minutes at the "Intense" setting.

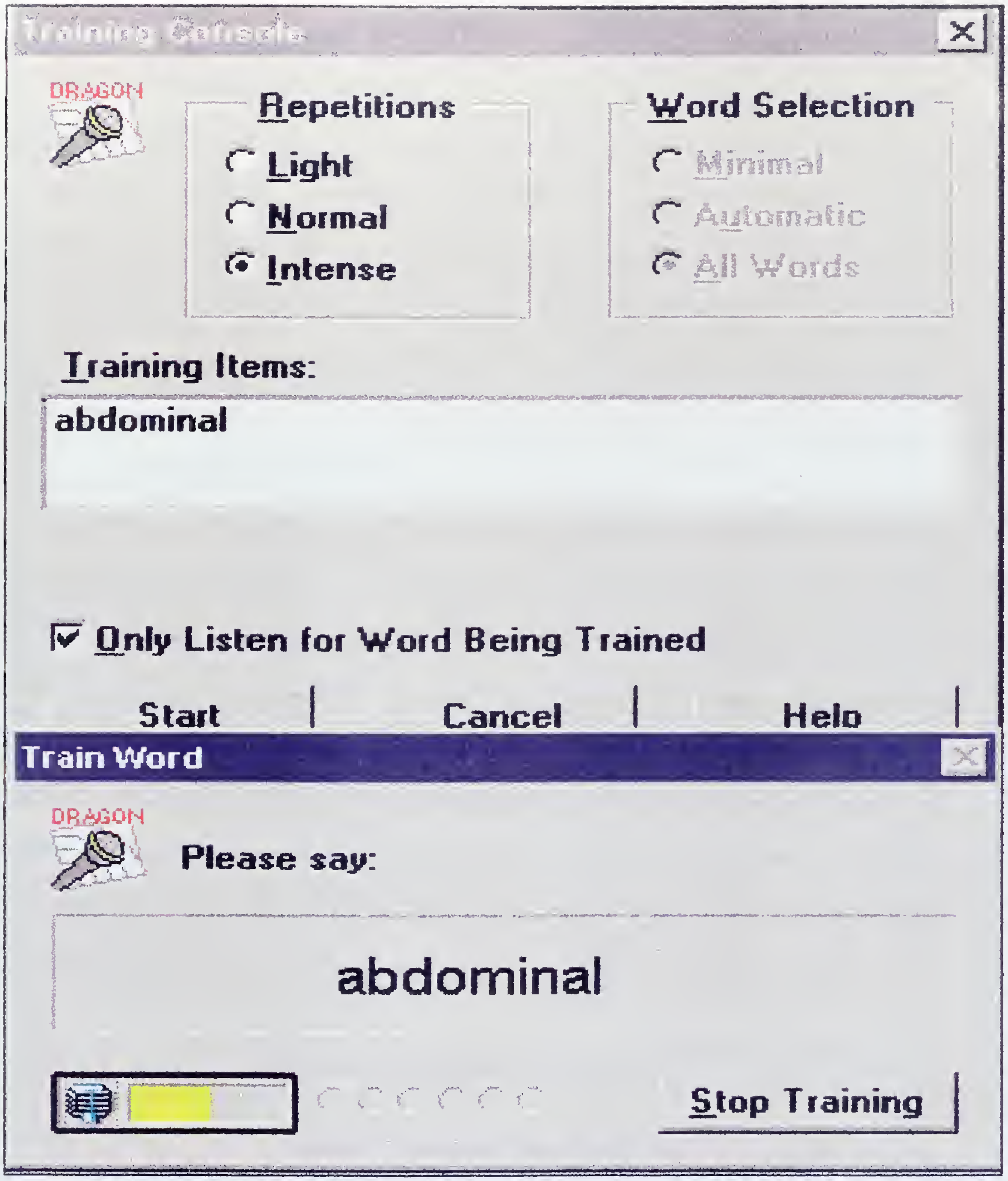
Quick Training involves training four groups of vocabulary types. These groups are "Correction Words," "Common Commands," "Dictation Words," and Additional Words. All four groups are recommended to be trained but need not be completed in one sitting. The Quick Training session can be started, stopped, and restarted when necessary. Completed training is never lost once it has been done, and training is always picked up where you previously left off. During training DragonDictate constantly adapts to your speech. This enables DragonDictate to constantly adjust the number of words required to be trained within each group. Thus, as training progresses DragonDictate will adjust the number of "Common Commands" required to be trained. This is why you may see the

count of words to be trained decreasing during the training session. Once training has been completed, DragonDictate is ready for use with any application that is Windows compatible.

3. Using DragonDictate

Before using DragonDictate, it is necessary to make sure that the microphone is properly adjusted for giving commands. The microphone should be situated about two inches away from the corner of the mouth of the user [Ref. 2: p. 22]. A headset microphone is recommended, optimally one of the three brands listed in the microphone selection dialogue box.

To begin using DragonDictate you must make sure that the microphone/headset is turned on by ensuring that the microphone window on the voicebar is either gray or yellow. The gray color indicates that DragonDictate is in a waiting mode (asleep), and the yellow color indicates that DragonDictate is ready and listening for a command. After ensuring that the microphone is turned on the user may begin to utilize DragonDictate to navigate Windows applications or to dictate into Windows compatible word processing applications.



Training Console

Repetitions
○ Light
○ Normal
● Intense

Word Selection
○ Minimal
○ Automatic
● All Words

Training Items:
abdominal

☑ Only Listen for Word Being Trained

Train Word

Please say:
abdominal

Figure 5. Training console

B. EVALUATION OF DRAGONDICTATE

The evaluation of DragonDictate was done in two stages. The first stage evaluated the dictation accuracy and learning capability of DragonDictate. The second stage evaluated the ease of navigation performed while working in Windows. The navigational ability of DragonDictate was evaluated by noting how well the software was able to accommodate opening and closing various Windows applications. The Windows applications used were Lotus 1-2-3, WordPerfect, MatLab, Netscape, Eudora, and Windows Program Manager.

1. Dictation Evaluation

The dictation and learning capability of DragonDictate were measured by dictating a standard passage consisting of 313 dictation words and commands into WordPerfect using DragonDictate. The passage was dictated six times, recording the number of mistakes, correcting the mistakes as they occurred (using the technique described in the DragonDictate User's Guide [Ref. 1, pp. 20-28]), and the length of time required to complete the dictation. The errors were calculated as a fraction of the total number of commands to give a percentage of each error type as well as the total amount of errors. For this study there were four types of mistakes that could be measured, which are listed and described below:

1. Type 1 - The software recognizes the wrong word or command but the correct word or command is located in the choice list.
2. Type 2 - The software recognizes the wrong word or command but the correct word or command is <u>not</u> located in the choice list.
3. Type 3 - The software heard nothing even though a word or command was uttered.
4. Type 4 - The software heard the correct word or command but performed the wrong action or did nothing.

These measures of performance were taken against the passage in Appendix D, which was dictated into WordPerfect. The results are depicted in Figures 6 and 7.

Figure 6 shows that with each trial, the number of errors made by DragonDictate decreased. The number of type 2 errors decreased with each trial due to those words not previously listed in the choice menu becoming candidates within the selections listed in the choice list. Eventually these words became recognized as the primary, or first selection, choices in the list. This means that they became the words that were recognized by DragonDictate as the input words uttered by the user. The other Error types became less frequent also, thus contributing to the improvement on the overall errors performed by the software.

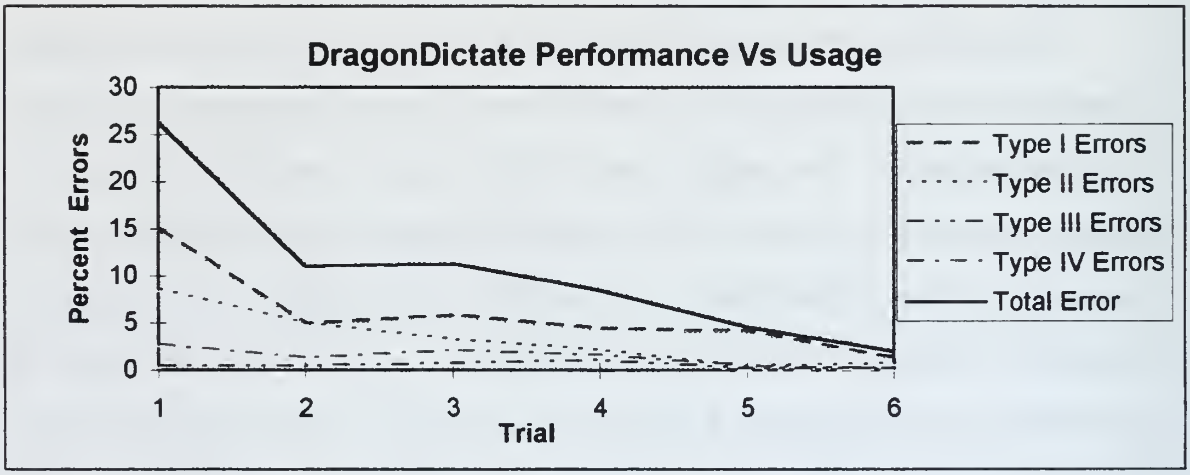


Figure 6. Number of recognition errors performed Vs Trials

Figure 7 demonstrates that with each use DragonDictate generally improved in its accuracy[1]. This supports Dragon Systems, Inc.'s claim that DragonDictate performance improves with usage. The greatest degree of accuracy reached during this evaluation was 98.03%. This was achieved within a controlled environment where the user was able to control the level of background noise. During this evaluation there was very little to absolutely no background noise present. With some background noise (maintenance man drilling in the adjacent room with the door closed) DragonDictate achieved an accuracy of 93.5%.

[1] Accuracy is defined as the complement of the total percentage of errors. It is 100 - the % errors.

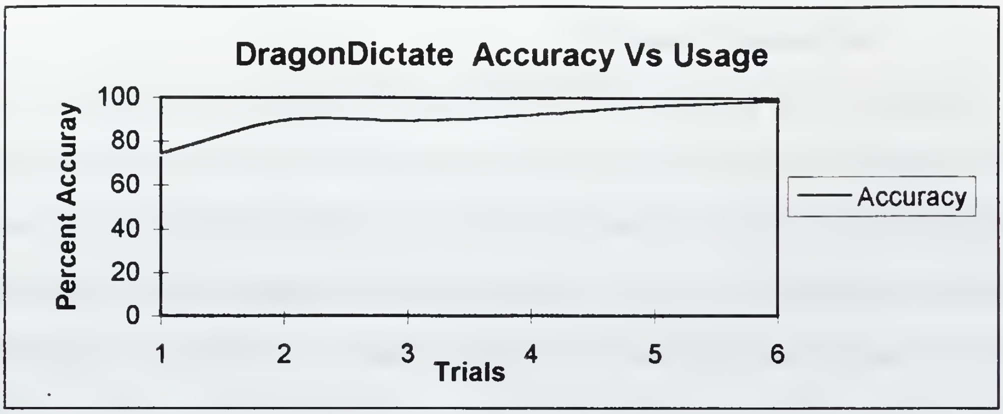


Figure 7. DragonDictate Accuracy Vs Trials

Along with the improvement of accuracy, the amount of time required to dictate the control passage decreased (Figure 8). As shown, with each successive use of DragonDictate, the length of time required to input the control passage was reduced. This was due to the improved level of accuracy. As accuracy improved, the user was able to increase the speed at which he dictated the text. Less time was expended correcting errors performed by the software. The longest input time was 20:35 (mm:ss) with an accuracy of 72.73%, the fastest input time was 9:45 with an accuracy of 98.03%.

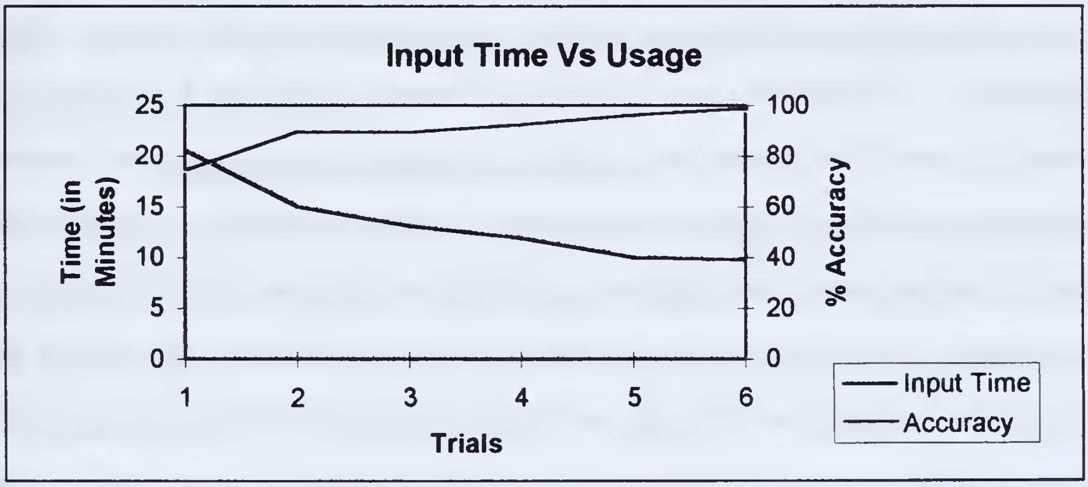


Figure 8. Accuracy Vs. Input time

2. Navigation Evaluation

Navigation with DragonDictate was flawless. All that was required to ensure reliable navigation was that the program being controlled by voice was properly added to the DragonDictate program group in windows. The technique for doing this is described in the User's Guide [Ref. 1: p. 18]. It is also necessary to ensure that the program being controlled is placed within the group and properly named. For example, Wordperfect 6.1 need only be named Wordperfect, while Lotus 1-2-3 may still be named Lotus 1-2-3. Other non-supported programs may still be controlled by training the name of the program. For example, Matlab is not supported and therefore is not part of DragonDictate's vocabulary. It is therefore required that the user train this particular word in order to start the program by voice. However, it is not necessary to train any of the commands within the menus of non-supported programs. DragonDictate is capable of tracking all of the commands within the menu and many of the button controlled commands as well.

C. SUMMARY

DragonDictate performed very well as an input device for the Windows operating environment. As a dictation input into word processing software and in conjunction with Matlab it proved to be outstanding. After some continuous use the software was able to adapt to the user's speech patterns and was able to improve accuracy to 98.03% within a quiet test environment. DragonDictate maintained an accuracy of over 90% in a noisy environment. The noisy environment was caused by a maintenance man drilling into a wall adjacent to the lab in which the evaluation was being performed. As a navigational input for Windows it performed equally well, though more work was required by the user in order to ensure that non-supported program applications were able to be initiated by voice. This procedure is described in detail in Appendix C.

IV. NAVIGATION SOFTWARE

Voice navigation software is basically a command and control type of application, as previously explained in chapter II of this paper. It allows the user to open, close, and to perform many menu driven commands within specific applications. The two navigation software packages implemented and evaluated for this study are Microsoft's Voice Pilot 2.0 - a part of the Windows Sound System software package, and Command Corp.'s IN ^{3} Voice Command For SPARCstation. The latter will be installed on a SPARCstation running Sun OS 4.1.3.

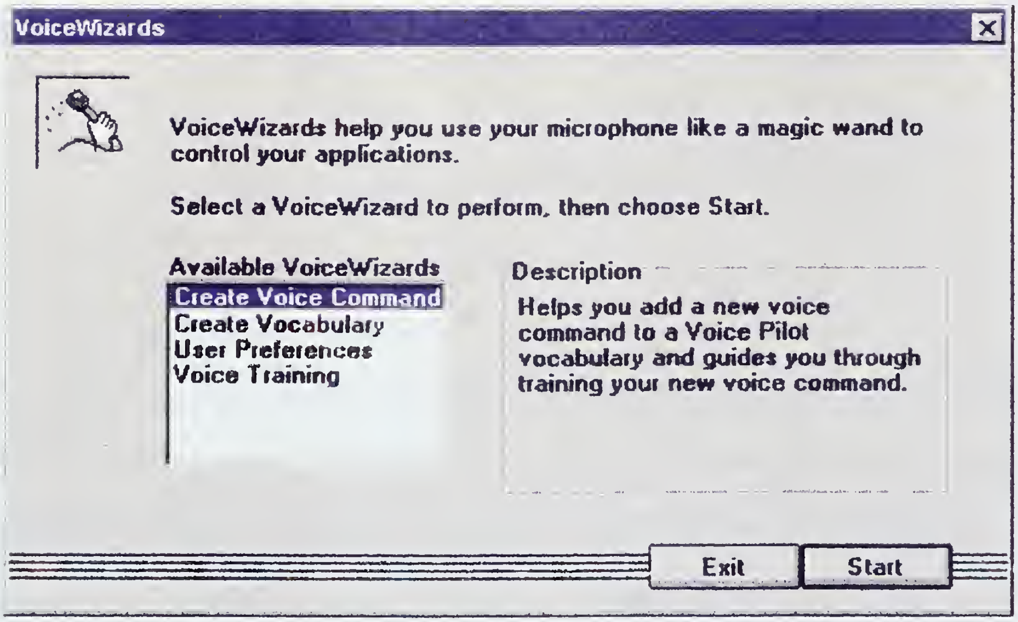
A. MICROSOFT WINDOWS VOICE PILOT 2.0

Voice Pilot works with the Microsoft Windows 3.x operating systems. It is compatible with *all* MS Windows compatible applications. Once installed, the application is fairly easy to use. It comes with several "wizards" - macros that automate or simplify the setup or usage of an application, which enhance its simplicity (Figure 9). These macros aid in the creation voice commands, new vocabularies, setting user preferences, and training voice commands.

1. Installing and Setting Up Voice Pilot

Implementation of Voice Pilot is quite easy. To install Voice Pilot, simply insert the diskette into the drive and using program manager click File|Run, then type A:\setup.exe, where "A" is the letter of the drive that the diskette is in, and simply follow the onscreen directions. The program requires a minimum of 10 megabytes of free hard disk space and about 2 megabytes of RAM and the following system requirements [Ref. 3: p. ix] :

1. An 8 or 16 bit Sound Blaster™ compatible sound card.
2. Microsoft Windows operating system version 3.1 or later.
3. An 80386SX or better IBM® compatible PC operating at 25 Mhz or faster.
4. A Microphone/headset.



VoiceWizards help you use your microphone like a magic wand to control your applications.

Select a VoiceWizard to perform, then choose Start.

Available VoiceWizards

Create Voice Command
Create Vocabulary
User Preferences
Voice Training

Description

Helps you add a new voice command to a Voice Pilot vocabulary and guides you through training your new voice command.

Figure 9. Voice Pilot Voice Wizards

During the initial setup the user is given the chance to set up a "Switch To" group that is used to store the application programs that the user wishes to navigate using voice commands. This group appears on the desktop as another application group except that it has the user's name as the title (Figure 10). This is a very important feature. The "Switch To" group name must match exactly. This "name matching" is how the program knows which user is associated with each specific "Switch To" group. The process for adding and removing applications within the "Switch To" group is the same as that required to add new program items the DragonDictate program group as described in Chapter III.

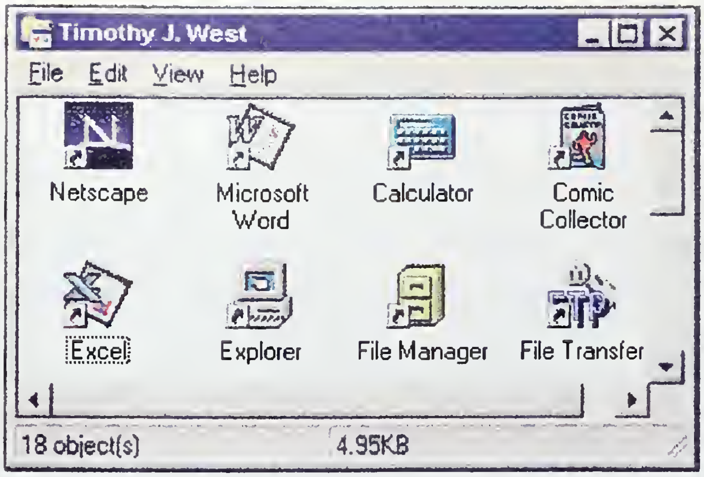
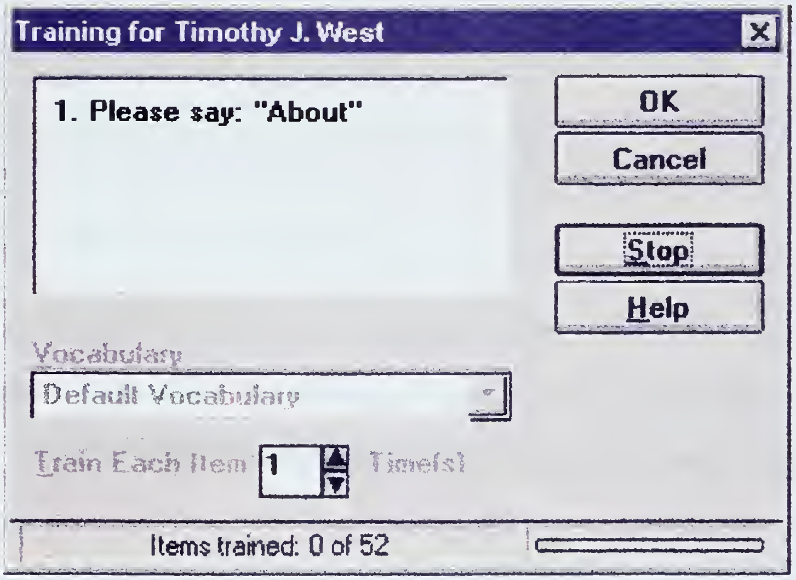


Figure 10. Switch To group

2. Training Voice Pilot

After completing the installation and creating the "Switch To" group, Voice Pilot is ready for voice training. Before any commands will be recognized voice training must be completed. The program includes a default vocabulary that is automatically loaded and trained at the beginning of voice training. Voice training is easily completed by using the voice training wizard included in the software. The training window (Figure 11) is very similar to the training window in DragonDictate, and it is very easy to use.


Training for Timothy J. West

1. Please say: "About"

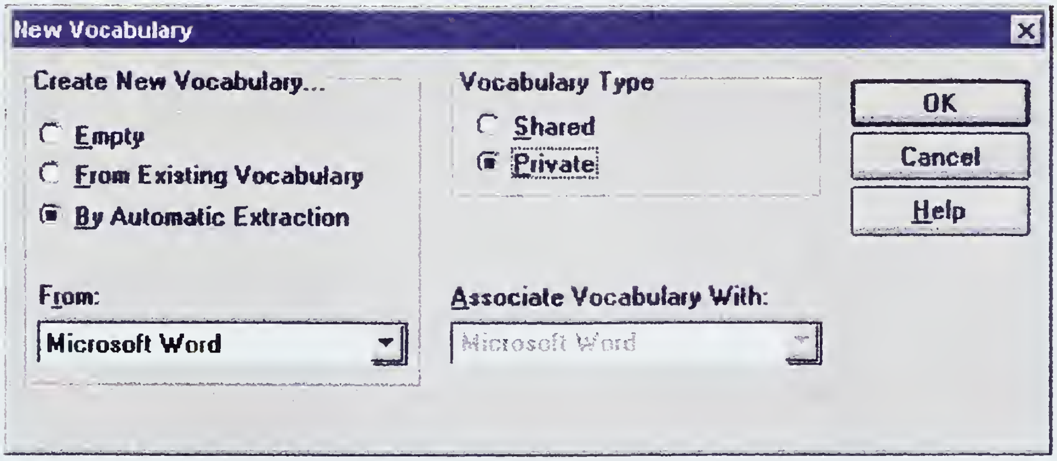
Vocabulary: Default Vocabulary

Train Each Item 1 Time(s)

Items trained: 0 of 52

Figure 11. Training Window

Once you have completed voice training, you can create new vocabularies by opening applications and then maximizing or opening Voice Pilot while the application is still opened. Voice Pilot allows you to create vocabularies by automatically extracting them from the open application (Figure 12). Voice Pilot will automatically create the vocabulary from all available menu items within the target application. Voice Pilot will select commands as far down as three or four levels of menu items. It also allows you to make any particular vocabulary a shared vocabulary or a private vocabulary. A shared vocabulary is available for use by any and all users that have access to Voice Pilot. A private vocabulary is only available to the user that created the particular vocabulary. Voice Pilot will then notify the user that the new vocabulary contains untrained commands and will allow the user to immediately train those commands.



New Vocabulary

Create New Vocabulary...

○ Empty
○ From Existing Vocabulary
● By Automatic Extraction

From: Microsoft Word

Vocabulary Type

○ Shared
● Private

Associate Vocabulary With: Microsoft Word

Figure 12. Creating a new vocabulary

The user is allowed to select either “Quick training” or “Untrained words.” Quick training consists of 52 commands that are common to most Windows compatible programs, and are the same for all applications. “Untrained words” are nominally 72 commands available for a specific application that have not been trained. These commands are extracted automatically from the available menu commands of the application. The number of words for a selected group of applications are listed below in

Table 1. Times for "Quick Training" vocabulary were the same for all applications, 7 minutes 35 seconds. There were no errors.

Application	# of Words	Time (*mm:ss*)	# of errors[2]
MS Word	78	11:24	0
Eudora	38	5:50	2
WordPerfect	78	10:32	0
Program Manager	20	2:34	0

Table 1. Applications trained using "Untrained words" selection.

B. IN^3 VOICE COMMAND FOR SPARCSTATION

IN^3 Voice Command by Command Corp. works under all audio-equipped SPARCstations using the following operating systems [Ref. 15: p. 2.]:

1. OpenWindows 3.x
2. Solaris 2.x (Sun OS5.x) .
3. Solaris 1.x (Sun OS 4.1.2 or 4.1.3).
4. Sun OS 4.1.1. - disregard warning messages from 1d.so that libc.so.1.6 has an older revision than expected.

IN^3 speech recognition technology uses voice templates created for each command and stores them in a *lexicon*. When in recognition mode, the program compares the templates and matches them to the input data coming from the microphone [Ref. 15: p. 6]. The software performs these comparisons continuously and in real time. It is for this reason that it is important to create these templates in a quiet environment with a strong

[2] Errors were words that required re-training due to background noise or I/O errors. These words were identified to the user by VoicePilot.

voice signal. Such templates will normally be well-matched and correctly recognized in an environment with typical office noise.

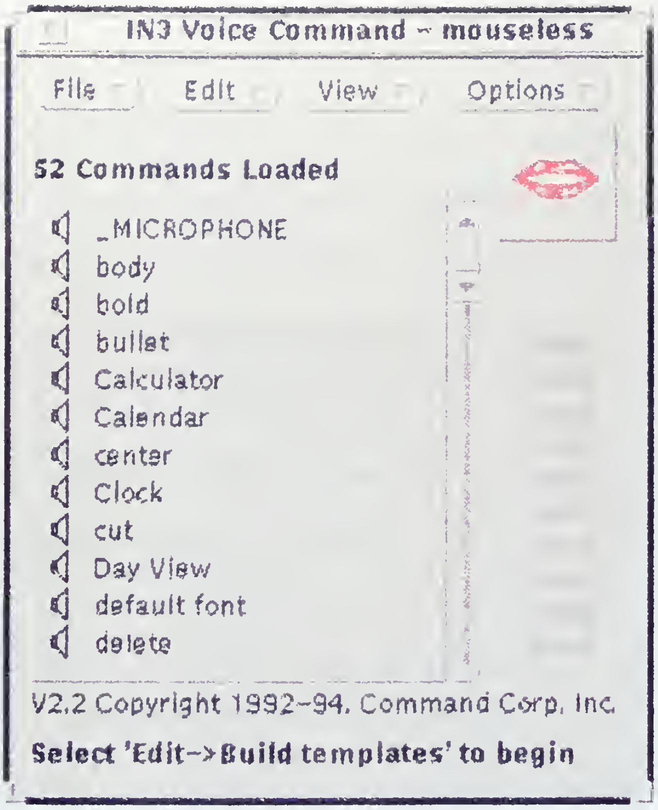
1. Installing IN^3 for SPARCstation

Installation of IN^3 for SPARCstation was performed by a network administrator. Installation on individual SPARCstations is fully described in the user's guide. The workstation itself should be audio-equipped. The workstation must have the necessary hardware and software installed to permit use of audio input and output. Upon completion of the installation IN^3 is ready for use.

2. Using IN^3 Voice Command

Once IN^3 is started by using the command "*in3*," the application performs a microphone check. The application requires that the user either opt to perform the microphone check, playback a sample (not available on initial use), or select continue. The "Mic Check" button allows the user to create a voice sample by saying the phrase "Sun Test." This is repeated again several times to allow IN^3 to adjust the microphone gain. This voice sample is then used as the playback sample. It is not necessary to playback the sample in order to begin using IN^3, but it is a good idea to play it back so that the user is able to hear the quality of the input being used as a template.

After completing the initial microphone check it is necessary to load a lexicon (set of commands) that becomes the active vocabulary. This is done by selecting "File" and then "Load Starter Lexicon." This opens a list of available lexicon files that are provided with the application. There are several to choose from and include lexicons for OpenWindows (openw.vcb), Frame Maker (framestart.vcb), and Vi (vi.vcb). Once loaded the list of available commands within the selected lexicon are displayed in the main window (Figure 13). The window shown lists templates that have not been created.



IN3 Voice Command – mouseless

52 Commands Loaded

_MICROPHONE
body
bold
bullet
Calculator
Calendar
center
Clock
cut
Day View
default font
delete

V2.2 Copyright 1992–94, Command Corp, Inc.

Select 'Edit->Build templates' to begin

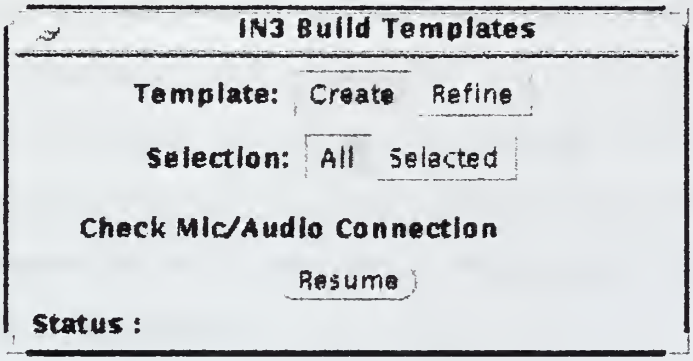
Figure 13. IN^3 Main window.

Many users will find that the lexicon sets provided do not provide the flexibility to go from one program or application to another without having to reload the proper lexicon. To solve this problem just include lexicons into the current lexicon. This increases the size of the lexicon, and thus increases the size of the available vocabulary. This is a feature unique to the SPARCstation version of IN^3, and allows the user to have just a single lexicon of unlimited size. The PC version does not allow the use of just one large lexicon. The limit is due to the memory requirements and processing limitations of the PC.

3. Building Templates

Once the user has loaded the desired lexicon(s), it is necessary to train the commands (build templates). Building templates is quite simple. The user must select "Edit," and then "Build Templates" from the IN^3 window menu. Once the "build Templates" dialogue is started, IN^3 will set either the "All" or "Selected" mode (Figure

14) depending on whether or not there are templates in the lexicon that are already trained. If templates exist then the “Selected” mode is set. If no templates are available then the “All” mode is set. In the “Selected” mode only those templates that require training are created. The user must then select “Create” to train templates.



IN3 Build Templates

Template: Create Refine

Selection: All Selected

Check Mic/Audio Connection

Status :

Figure 14. Template creation in "All" mode.

IN^3 begins to train (create) templates when the “Begin” button is selected. The “Begin” button then becomes a “Pause/Resume” button. IN^3 then begins the training (creation) dialogue. The user is asked to say commands (templates) several times, usually twice, unless a command is not recognized or there is a problem with the microphone input. IN^3 senses input deficiencies and then notifies the user. Should the user pause too long between utterances, IN^3 detects this as an error and notifies the user of input deficiencies. The user is then allowed to correct the error or continue training. After training each command, IN^3 then goes through the entire list of words/commands trained and asks the user to repeat them. IN^3 performs this routine to ensure proper training has occurred. It also catches any errors made during training and retrains the command at this time. The entire process for 116 commands took only 10 minutes and 33 seconds.

4. Adding And Editing Commands

Editing, adding, and modification of commands is performed in the Edit Commands window (Figure 15). This window allows the user to modify, delete, or reset the selected command and also to add any new commands. When adding a new

command the user begins by clicking the "Edit" menu choice, then selecting the "Edit Command" menu selection. This starts the "Edit Command" dialogue window. The user must then either type in the name of the new command or select a command listed in the IN^3 main window. The complete method for adding and editing commands are given in the user's guide. The specific key combination or mouse movements and button presses/clicks are programmed into the "Keystrokes:" box by using the "Window/Pointer Probes:" macro. By selecting "Names" and then changing the focus to the desired application window and clicking the left mouse button, the user is able to capture the name of the application. By the "Packing" button, the user is able to track (capture) the mouse movements and button clicks.

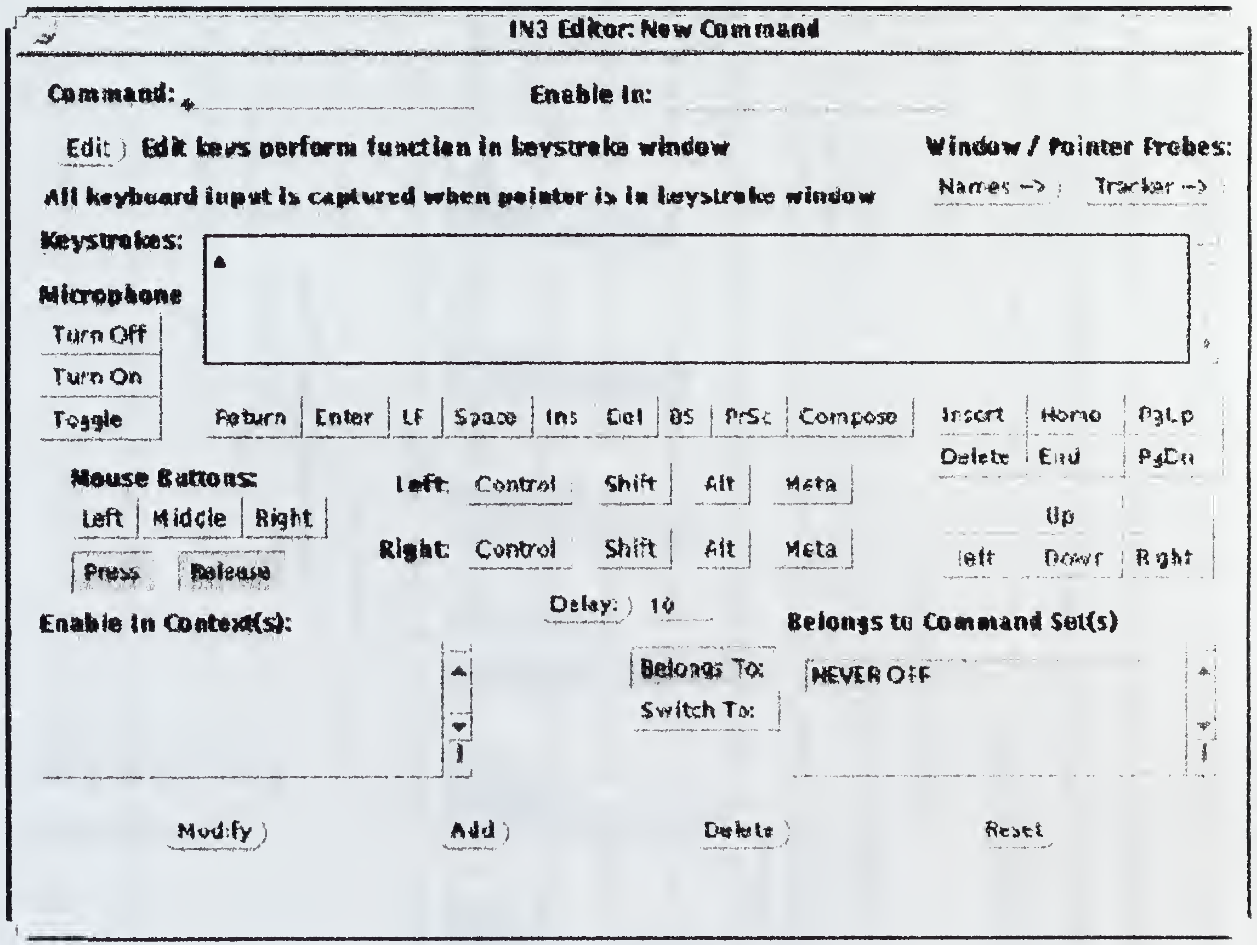
IN^3 is aware of which commands may be executed with each particular application, or X-aware as described by Command Corp. This allows the user to build a single large lexicon containing an unlimited number of templates to control the most used functions and applications by voice without having to switch between lexicons. Thus it is possible to have all voice commands needed by the user located in one vocabulary file. The user simply continues to add commands and templates to his lexicon.

IN^3 controls application startup in one of three methods: using a windows management mode, an application execution mode, or by using embedded commands. In windows management mode the command is preceded by "*f.wmm*" and then the command is typed. For example, to start the Shell tool the user would type "*f.wmm shelltool {CR}*" in the "Keystrokes:" box in the "Edit Command" dialogue window. The windows management mode execution method will startup the Shell tool by 1) maximizing the shelltool if it is currently running as an icon, 2) bringing the shelltool to the front if it is running but is hidden under other open windows, or 3) starting the shelltool application if the shelltool application is not currently running.

Using the application execution mode, IN^3 will start a new instance of the application even if the application is currently running. In this mode the command is preceded by "*f.exec*" and then the command is typed. Using the previous Shell tool example, in this mode the command would be "*f.exec shelltool {CR}*". This command would start a brand new instance of shelltool if the application were running or not running.

Using embedded and conditional commands allows the user to specify the conditions under which an application is started. IN^3 has 15 recognized embedded commands and two conditional commands ("True" and "False") that can be used together in many different combinations. This allows the user to have greater flexibility and control navigating applications. For example, {*Front:t:/usr/spool/mail*} is used to bring the Mailtool window titled "/usr/spool/mail" to the front of the display screen and to give that window the focus for command execution. This is very useful when using applications that utilize multiple windows, such as Mailtool. Mailtool opens multiple windows to view or to compose mail. Using the "Front" example above allows the user to execute commands in those windows without having to use the mouse to switch the focus to that particular window.

Using the conditional commands adds even greater functionality to the voice commands. The combination {*Front:all:bob*}{*False:open:bob*}{*False:exec:cmdtool -name bob*} does three things. First it attempts to bring forward a window called "bob." If that fails it tries to open an iconified window named "bob." Should that fail, it starts the Commandtool and tells it to use the name "bob" as its resource name.



Edit keys perform function in keystroke window

All keyboard input is captured when pointer is in keystroke window

Delay: 10

Belongs To: NEVER OFF

Figure 15. Edit Command dialogue window

C. EVALUATION

The evaluation of both IN^3 and VoicePilot consisted of giving navigational commands and taking note of all errors that occurred. Usability and ease of training the vocabulary and adding commands, were also taken into consideration while evaluating both software packages.

1. VoicePilot Version 2.0

VoicePilot performed reasonably well in a moderately quiet environment. Moderately quiet means in this case that the environment was less quiet than that of a normal office. In this environment the navigational ability of VoicePilot was nowhere close to the level of accuracy that would be required in a noisy shipboard environment. Figure 16 depicts the range of accuracy of VoicePilot over a period of six trials. Using 114 trained commands within supported programs (MS Word, WordPerfect, and Program Manager), VoicePilot was evaluated by actually navigating the supported Windows applications. The maximum accuracy reached by VoicePilot was 77.77%. Most users would probably desire a minimum of 90% accuracy. Any less than that and it would be easier to do navigation by hand.

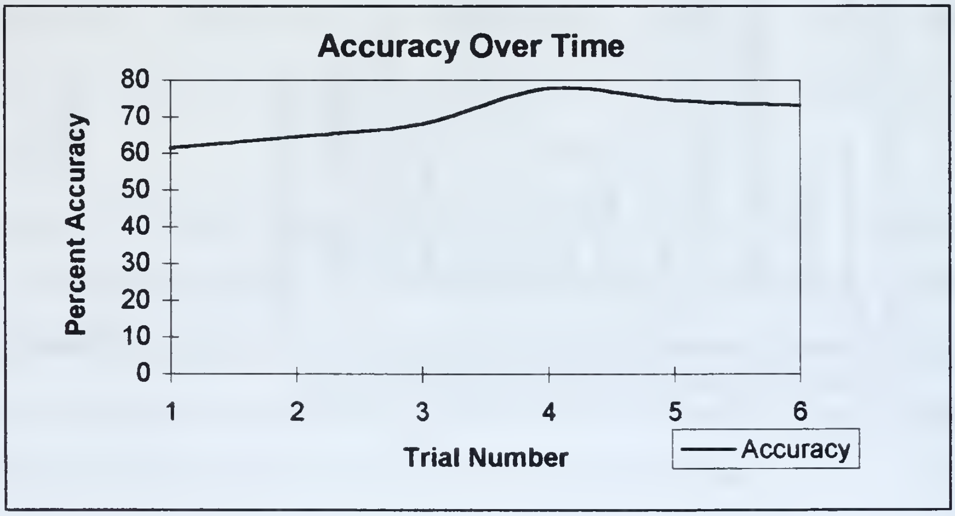


Figure 16. VoicePilot accuracy

The errors made by VoicePilot were categorized into three types: 1) commands that were unrecognized or not heard by VoicePilot, 2) commands that were unable to be corrected within the VoicePilot correction dialogue, and 3) commands that were incorrectly recognized by VoicePilot which resulted in unwanted actions being performed by the software. The percentage of these type errors as a part of the overall amount of errors is shown in Figure 17.

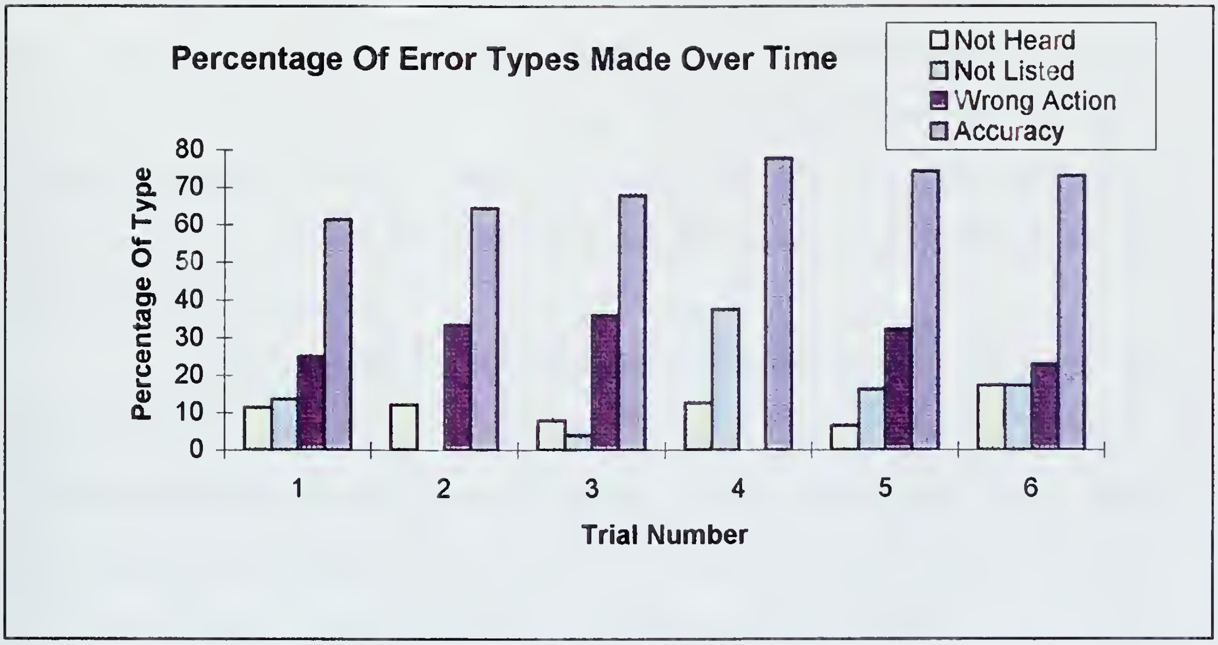


Figure 17. Error type percentages

The number of errors made by VoicePilot that resulted in some unwanted action was very high, as shown in Figure 17. Though no major setbacks were experienced, the potential for disaster is quite extreme. Although the majority of the errors were corrected, many of the commands were not able to be corrected using VoicePilots correction dialogue window. There appears to be no true pattern of improvement. The same commands can be incorrectly recognized time after time, even with corrections being made. Even then the same errors still occur and sometimes the word needing to be replaced for what VoicePilot recognized is not listed among the choices of commands.

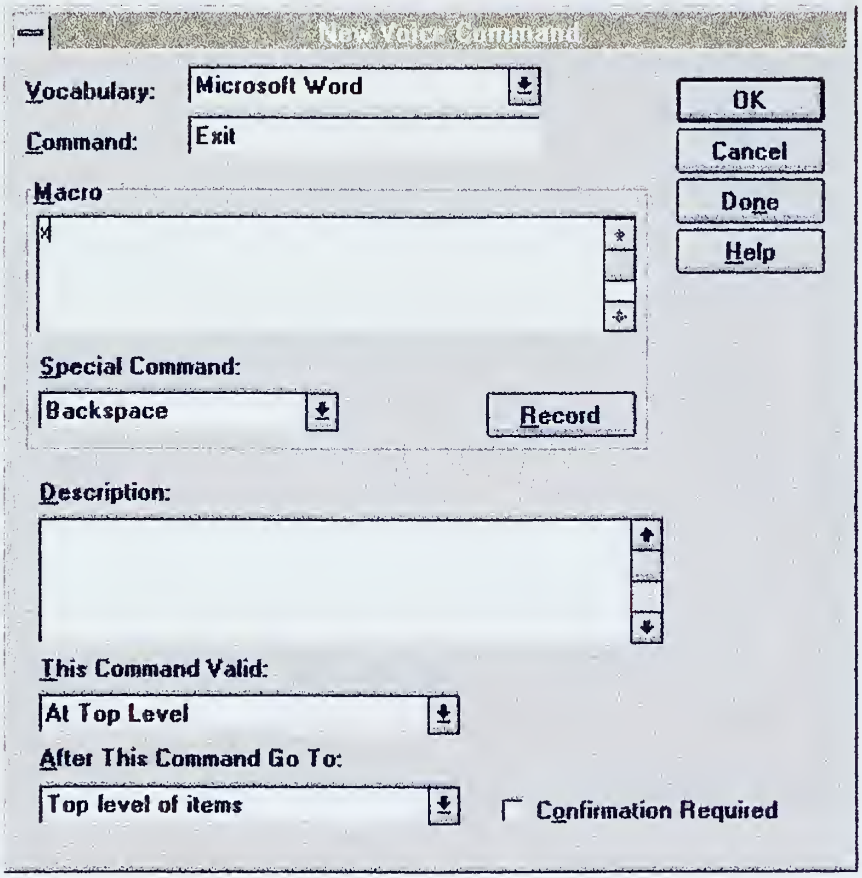
a. *Adding Vocabularies and Commands*

Adding new vocabularies was simple and quick with VoicePilot. All the user needed to do was to open the application for which the new vocabulary was to be used and then open VoicePilot. After opening VoicePilot, the user needed to choose the menu item "Vocabulary" and then choose "New Vocabulary." Once in this dialogue the user need only choose the target application and to check the radio button for adding the new vocabulary by automatic extraction (Figure 12). VoicePilot then extracts the vocabulary from the menu items of the target application and then offers to allow the user to conduct training for the new vocabulary of commands. The new vocabulary will be opened automatically by VoicePilot any time that the associated application is started while VoicePilot is active.

Adding individual voice commands is a different series of operations. In order to do this the user chooses "New Voice Command" from the "Vocabulary" menu. VoicePilot then opens the "Add New Command" dialogue window (Figure 18). The User then selects the application for which the new command is to be associated, the name of the new command, and the keystrokes associated with the command that are to be replace. This is a very easy way of creating a new command, though being able to record the mouse movements and then substituting them with the voice would probably be much easier. Not every user is going to be familiar enough with every application to know exactly which keystrokes perform which function. Most functions are easily accomplished by pressing a button on a toolbar with the mouse. The user must then train the new command in order for it to be recognized by VoicePilot.

b. Ease of Use

VoicePilots interfaces made the program extremely "user-friendly" that is, the program was not very hard for even the novice computer user to operate. The many "wizards" included with the program made training and adding new vocabularies even simpler. The "User Preferences" wizard enabled the program to optimize its settings just by asking the user to say nine phrases (standard phrases that were the same each time the wizard was used) into the microphone/headset. The user never had to worry about manually setting any sound card settings or voice input levels. Though there is a manual setting choice, it was never used. The software will alert the user if the automatic setting was not able to be set and would then instruct the user to manually set the device input level.


New Voice Command

Vocabulary: Microsoft Word

Command: Exit

Macro

Special Command: Backspace

Description:

This Command Valid: At Top Level

After This Command Go To: Top level of items

☐ Confirmation Required

Figure 18. Add New Command window

2. IN^3 Voice Command

IN^3 Voice Command performed very well under identical environmental conditions as VoicePilot. IN^3 Voice Command was installed and operated on a SPARCstation using SunOS 4.1.3 and OpenWindows version 3. An Audio-Technica MT858 microphone was used as an input device. The microphone was very sensitive and could pick up the low pitched whine of the CPU cooling fan inside the SPARCstation. The user was able to position the microphone up to two feet away and still have a good input signal for the operation of IN^3.

114 commands using the vocabulary listed in Appendix B were used to evaluate IN^3. The accuracy of IN^3 was very poor during the initial use of the application. With continued correction of errors and refinement of the voice commands, the accuracy of IN^3 was able to be improved to 90.91%. Figure 19 shows the progressive improvement of accuracy with each use of IN^3. Most users would feel very comfortable using IN^3 at 90% or better. With increased use and refinement, the accuracy of IN^3 should be able to be improved to well over 90%.

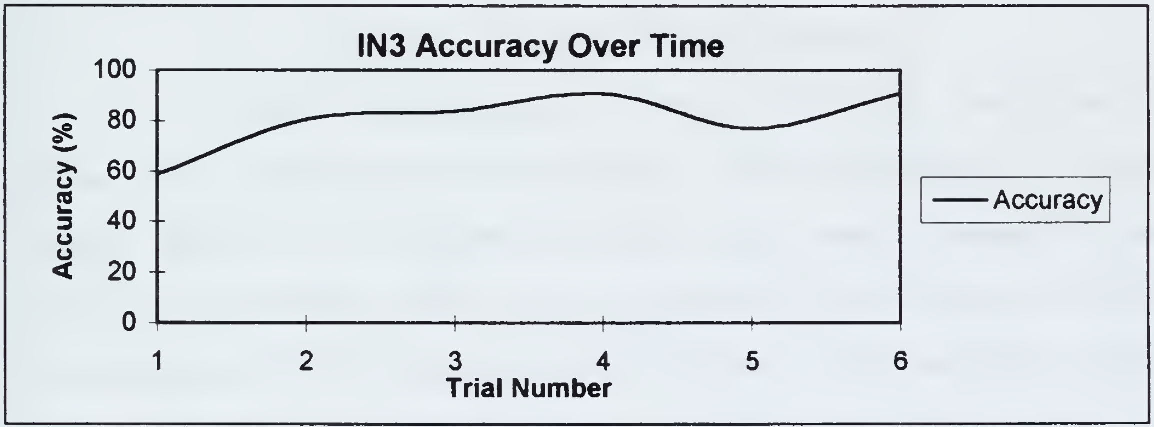


Figure 19. IN^3 Accuracy over time

The errors made by IN^3 were able to be categorized into three types of mistakes: 1) the command was not heard or recognized by IN^3, 2) the command was recognized but there was no action performed by the software, or 3) the command was recognized but the

wrong action was taken by the software.[3] As depicted in the chart in Figure 20, even in times of great accuracy the number of errors that resulted in some unwanted action was high. Though most of the unwanted actions were of a benign nature and were easily corrected by resetting the movements or modifying the command to perform the correct actions, the consequences of these unwanted actions could potentially be disastrous.

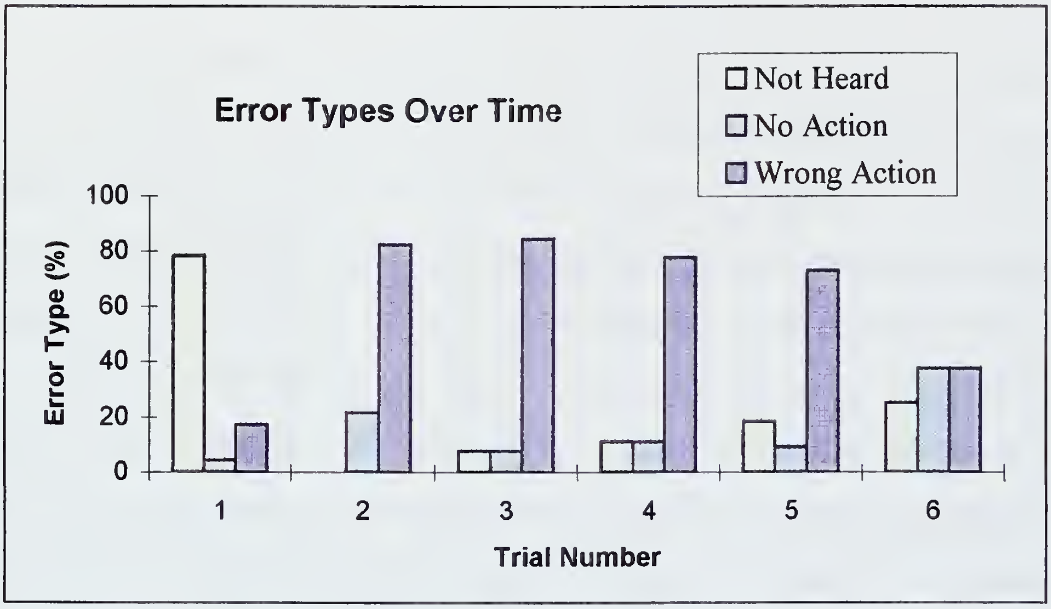


Figure 20. Error Type[4] percentages committed by IN^3

a. Adding New Vocabularies and Commands

Adding new vocabularies in IN^3 were as simple as just opening the "File" menu selection and choosing the "Add lexicon," "Add starter lexicon," or "Include lexicon" selections. The "Add lexicon" selection adds a template located in the users directory. This template could be one of several that the user may have created or modified from the lexicons included with the program. The "Load starter lexicon" selection allows the user to select and load any one of the nine included lexicons. The difference between these lexicons and those that are added by the "Add lexicon" selection

[3] Wrong action included bringing up the wrong application window and/or executing improper mouse movements or clicks.

[4] Error Types depicted are a percentage of the total number of errors committed by IN^3

is that these starter lexicons are not yet trained, and those loaded using the "Add lexicon" selection may or may not be trained. The "Include lexicon" selection allows the user to add vocabulary commands from different lexicons into one large lexicon, creating one large vocabulary file. The advantage of doing this is that the user will not have to switch templates when different applications are started or selected for use.

Adding individual commands is done using the "Edit Command" dialogue as previously described. Learning how to use embedded commands, capturing keystrokes, and enabling commands to operate within specific applications is the tricky part. Learning the use of embedded commands is almost like learning a new programming language. The examples given in the User's Guide are not very clear, and the User's Guide itself reads more like a technical manual than a guide. It is extremely helpful if the user has some general or basic knowledge of UNIX or OpenWindows. Several calls were made to Command Corp. for technical help on how to program some of the commands, especially commands dealing with applications using multiple windows. The result of the technical help was the use of the "Front" command previously described. This technique is described in the *IN Cube Voice Command for SPARCstation version 2.2.2 Release Notes* that are installed in the *usr/lib/in3/info/* directory in the file "relnotes.ps". This document contains notes, changes, and corrections to the documentation included in the package with the software.

D. SUMMARY

In this chapter we have looked at the two navigational software packages evaluated in this study, VoicePilot and IN^3 Voice Command. We have seen that both were produced to perform the same type of operations, that is to navigate between applications in a windows environment. As navigational input devices for windows operating system environment, VoicePilot was found to be less than desirable due to its low accuracy. In contrast, IN^3 performed well as a navigational device, reaching a 90.91

% accuracy rate after continuous use. IN^3 and VoicePilot showed that there is a propensity for both packages to perform unwanted actions when there is an error made in the recognition of a command. This is not an attribute that any user would want. In this study the unwanted actions were benign, but the consequences of such error types in other situations could be potentially disastrous.

V. CONCLUSIONS AND RECOMMENDATIONS

A. CONCLUSIONS

In the past few years the DoD has placed an emphasis on Command, Control, Communications, Computers, and Intelligence For the Warrior (C4IFTW). C4I is the future for all the military services, and is playing a major role in the planning of future capabilities, makeup, and budgetary issues within DoD. A major factor in C4I FTW is the interface between man and computer. One of the technologies which is "coming of age" is voice recognition. Within a few years (some experts say within the next ten years) giving "orders" or inputting data into a computer by voice may be the normal way of doing business. For C4IFTW, to give the computer a common look and feel so that interfacing with it is almost natural, one solution is to incorporate voice recognition as an interface between the user and the machine.

Voice technology has made great strides within the past three to five years. Manufacturers are beginning to produce voice recognition packages that are ready to use right out of the box. Training commands and vocabularies is optional. These voice recognition packages are being produced to support all of the major computing platform operating systems. These include MS Windows (version 3.x, 95, and NT), UNIX, SunOS, OpenWindows 3.x, and even OS/2. With more of the computing industry focusing on multimedia, voice recognition is becoming a more popular technology.

This thesis took a look at three voice recognition software packages currently on available in the commercial market, DragonDictate version 1.3, VoicePilot version 2.0, and IN^3 Voice Command for the SPARCstation version 2.2.2. These three packages were implemented on various systems and evaluated. Of these three packages DragonDictate was the best choice for dictation and navigation. It was shown that DragonDictate's accuracy improved steadily with increased usage, maintaining an accuracy above 98 % in a quiet environment, and 93.5 % accuracy in a relatively noisy environment. The accuracy was able to improve because DragonDictate was able to "learn" the users speech patterns, and apply corrections to voice commands to avoid future errors. The user needed to perform a twenty minute initial training, but this was the only extensive training the program needed. Navigational commands were not required to be trained for each specified application. VoicePilot and IN^3 Voice Command both required training for each application or command within each vocabulary. DragonDictate was the simplest package to use, as well as the most accurate in recognizing voice commands.

B. RECOMMENDATIONS FOR FURTHER RESEARCH

This thesis provides a preliminary study on the application of voice recognition technology. Following is a list of three areas dealing with applications of voice recognition technology (although this is clearly not an exhaustive list of possible research areas involving voice recognition).

1. Voice Recognition and the Internet

This thesis used voice recognition to automate many of the menu and button commands involved with software to access the Internet and the World Wide Web such as Netscape, Mosaic, and FTP tools. However, once connected many of the functions performed while "browsing" the Web were still done using the mouse. Possible research topics exists in the area of SLAM (Spoken Language Access to Multimedia)[5] and its possible implementation on a machine at the Naval Postgraduate School.

2. Service Area Specific Applications Of Voice Recognition

Use of voice recognition in many commercial professional areas has become popular. Research topics can be examined in the possible application of using profession

[5] Spoken Language Access to Multimedia (SLAM) is a spoken language extension to the graphical user interface of the World-Wide Web browser Mosaic being developed by the Center for Spoken Language Understanding (CSLU) at the Oregon Graduate Institute. SLAM uses the complementary modalities of spoken language and direct manipulation to improve the interface to the vast variety of information available on the Internet.... SLAM is believed to be the first spoken-language interface to the World-Wide Web to be easily implemented across platforms. [Ref. 16]

specific voice recognition software in the military counterpart or equivalent Warfare Specialty area, especially under field conditions. Many vendors are currently shipping special editions of voice recognition with vocabularies specifically created for the medical and legal professions.

3. Use Of Voice Recognition Across Platforms

A group of people suffering from RSI (repetitive strain injury) have utilized a2x, a piece of public domain software designed to interface the DragonDictate speech recognition system on a PC to a workstation running the X window system. Research could be performed at Naval Postgraduate school to utilize a2x to interface voice recognition on a PC to a workstation.

APPENDIX A. DRAGONDICTATE VOCABULARY LIST

This is a sample list of the vocabulary words used in DragonDictate for Windows version 1.3. Bold typeface words in the "command" column are the spoken command words (what the user says to cause the performance of a specific action). The "actions" column lists a brief description of what each command does or the resulting action[6].

A. ALWAYS ACTIVE COMMANDS

These commands are available at all times.

Command	**Action**
Command Mode	Sets DragonDictate to Command and Control Mode
Dictate Mode	Sets Dragon Dictate to Dictation Mode
Go To Sleep	Sets DragonDictate to passive mode. The software is not listening for commands.
Wake Up	Sets DragonDictate to active listening mode. The software is listening for commands or dictation words.
What Can I Say?	Lists relevant vocabulary for the current application
Oops	Starts DragonDictate correction sequence

B. GLOBAL COMMANDS

These commands are available for use at all times except when 1) training a word, 2) the user has already uttered **Bring Up**, 3) during arrow or mouse movement, or 4) while DragonDictate is in the passive listening mode.

Command	**Action**
Bring Up	Starts an Application

[6] This convention using "command" and "action" columns is used consistently in all appendices to denote what is uttered and the resulting action.

Computer Please	Puts DragonDictate into temporary Command and Control mode during dictation.
Drop List	Shows the list in a ListBox.
Move Voicebar	Moves the position of the Voicebar from one of the four corners of the screen to the next in clockwise rotation.
Type Word	Begins the macro to allow for typing the word following the command.
Voice Menu	Switches to the Voice Menu dialogue.

C. WINDOWS COMMANDS

These commands control the attributes of the windows. Active window is used to denote the window possessing the focus of attention.

Command	**Action**
Clear Desktop	Clears the desktop of all open windows.
Close Window	Closes the currently active window
Maximize	Maximizes the currently active window, if not already maximized.
Minimize	Minimizes the currently active window, if not already minimized.
Move window	Grab focus of window and allows the window to be positioned by voice using the mouse commands.
Next Window	Switches focus to the next open window from the current window.
Previous Window	Switches focus to the previous open window from the current window
Restore	Restores attributes of the current window from any change that has occurred, such as minimizing, maximizing, or moving.
Size Window	Allows for the changing of the size of the window by dragging with the mouse using the mouse commands.
Window Menu	Opens the current menu of window options.

D. ARROW MOVEMENT

These commands control arrow movement. When the command is spoken the arrow begins to move as required. **Stop** ends the arrow movement.

Command	Action
Cancel	Cancels current action
Down	Moves arrow or mouse down
Faster	Increases the speed/rate of movement of the arrow
Left	Moves the arrow in the left direction.
Move Down	Moves arrow down, and also moves down a list in a list box.
Move Left	Moves arrow in the left direction
Move Right	Moves arrow in the right direction
Move Up	Moves arrow in the up direction
Move Down 1 … Move Down 5	Moves the arrow or selection in list box down 1-5 increments.
Move Up 1 … Move Up 5	Moves the arrow or selection in list box up 1-5 increments.
Right	Moves arrow in the right direction
Slower	Slows the speed/rate of arrow movement.
Stop	Stops arrow movement
Up	Moves arrow in the Up direction

E. DICTATION COMMANDS

These commands are used when dictating text.

Command	Action
Back 1 …Back 5	Moves cursor back one to five words by specified increment.
Begin Capitalize	Types all words with first letter in uppercase.
Begin Document	Starts the document. Indents 5 spaces to begin paragraph and will Capitalize the first letter of the next "dictation" word spoken.
Begin Lowercase	Types all words dictated in lowercase
Begin No Space	Prevents DragonDictate from placing a space between words spacebar
Begin Title	Causes DragonDictate to use title capitalization rules.
Begin Uppercase	Types all words letters in uppercase.
Bottom of Document	Takes the cursor to the bottom of the document

Capitalize Next	Capitalizes the First letter of the next word.
End Capitalize	Stops the actions taken by **Begin Capitalize**.
End Lowercase	Stops the actions taken by **Begin Lowercase**.
End No Space	Stops the actions taken by **Begin No Space**.
End Title	Stops the actions taken by **Begin Title**.
End Uppercase	Stops the actions taken by **Begin Uppercase**.
Lowercase Next	Types the next dictation word in lowercase.
New Line	Begins a new line of text. Does not begin the line in paragraph format.
New Paragraph	Starts a new paragraph. Indents first line and capitalizes the first word.
No Space	Suppresses automatic space between words.
Normal Case	Types words in normal sentence case
Scratch That	Deletes the last word dictated.
Scratch 2 ... Scratch 5	Deletes the number of words stated by the numeral, i.e., **Scratch 2** would delete the last two words dictated.[7]
Top of Document	Takes the cursor to the top of the document
Uppercase Next	Types the next diction word in all uppercase letters.

F. MOUSE MOVEMENT COMMANDS

These commands are used to control the mouse movements. By saying **Mouse** + ***the desired direction***, the mouse movement is initiated. Saying **Stop** ends the mouse movement.

Command	**Action**
Button Click	Clicks the Left mouse button
Button Double Click	Double Clicks the left mouse button.
Cancel	Stops current command
Double Click	Double Clicks the left mouse button.
Down	Moves mouse cursor down.
Drag Down	Drags object/Window in the down direction.
Drag Left	Drags object/Window in the left direction.
Drag Lower Left	Drags object/Window toward the lower left hand corner of the screen.
Drag Lower Right	Drags object/Window toward the lower Right corner of the screen.
Drag Right	Drags object/Window in the right direction

[7] These commands were programmed by the author. Details on how this was accomplished are in Appendix C.

Drag Up	Drags object/Window in the up direction
Drag Upper Left	Drags object/Window toward the upper left corner of the screen.
Drag Upper Right	Drags object/Window toward the upper right corner of the screen
Faster	Increases the speed/rate of mouse movement
Left	Moves mouse cursor in the left direction
Lower Left	Moves mouse cursor in the lower left direction
Lower Right	Moves mouse cursor in the lower right direction
Mouse Down	Same as **Down**
Mouse Left	Same as **Left**
Mouse Lower Left	Same as **Lower Left**
Mouse Lower Right	Same as **Lower Right**
Mouse Right	Moves mouse cursor in the right direction.
Mouse Up	Moves mouse cursor in the up direction.
Mouse Upper Left	Moves mouse cursor in the upper left direction.
Mouse Upper Right	Moves mouse cursor in the upper right direction.
Right	Same as **Mouse Right**
Right Button Click	Clicks the right mouse button
Slower	Slows the speed/rate of mouse movement
Stop	Stops mouse movement
Up	Same as **Mouse Up**
Upper Left	Same as **Mouse Upper Left**
Upper Right	Same as **Mouse Upper Right**

G. SYMBOLS AND PUNCTUATION

These commands are used to type these commonly used symbols and punctuation marks. This is just a partial listing. DragonDictate supports all of the ASCII symbols.

Command	**Action**
Ampersand	Types character “&”
Asterisk	Types character “*”
At Sign	Types character “@”
Caret	Types character “^”
Open Brace	Types character “{”
Close Brace	Types character “}”
Open Bracket	Types character “[”
Close Bracket	Types character “]”
Open Paren	Types character “(”

Close Paren	Types character “)”
Open Quote	Types character “ “ ”
Close Quote	Types character “ ” ”
Comma	Types character “,”
Dollar Sign	Types character “$”
Period	Types character “.”
Pound Sign	Types character “#”
Slash	Types character “/”
Backslash	Types character “\”
Pipe	Types character “|”
Tilde	Types character “~ “

H. CORRECTION COMMANDS

These Commands are used to correct errors.

Command	**Action**
Cancel	cancels current action
Choose 1 … Choose 10	Selects the numbered word from the list of possible words heard by DragonDictate and then returns the user to Dictate mode.
Edit 1 … Edit 10	Allows the user to edit the selected word. Also opens a list of words derived from or similar to selected word.
Modify Word	Allows the user to enter the modification dialog to change actions performed by the command.
OK	Ends the current action taken by the user and returns them to the Dictate or Command mode, whichever mode from which the action was initiated.
Oops	Starts DragonDictate correction sequence
Select 1 … Select 10	Selects the numbered word from the list of possible words heard by DragonDictate. Does not return user to dictate mode. Allows for multiple corrections.
Spell Mode	Allows user to spell word phonetically using Alpha-Bravo words and Listing possible words as they are spelled.
Word Left 1 …Word Left 5	Moves the Cursor left one to five words and lists possible alternatives.
Word Right 1 … Word Right 5	Moves the Cursor right one to five words and lists possible alternatives.

I. NUMBERS AND KEYS

Commands for numbers from 1 through 99 are given by just saying the number. The same is also true for keys on the keyboard. For example to use the {Tab} key the user would just say "**Tab**." The following list produces the rest of the number set.

Command	Action
Zero	Types character "0"
Hundred	Types characters "00"
Thousand	Types characters "000"
Million	Types characters "000000"
Point	Types character ".", without two spaces following as if used for punctuation.
Comma (numeric)	Types character "," no space following.

APPENDIX B. IN^3 EVALUATION VOCABULARY

The following is a list of the commands used for the evaluation of IN^3 Voice Command for SPARCstation version 2.2.2. Combinations of theses commands were used to comprise the 114 commands.

Command	**Action**
_MICROPHONE	Switches microphone off and on.
Audio Tool	Manages Audiotool using *f.wmm*[8] mode.
Binder	Manages Binder application using *f.wmm* mode.
Bookmarks	Presses the "Bookmarks" button in Netscape.
Calculator	Manages Calculator application using *f.wmm* mode.
Calendar	Manages Binder application using *f.wmm* mode.
Cancel	Presses the "Cancel" button in Frame Maker
Clear Window	Clears the editing window when composing mail in Mailtool.
Clock	Manages Clock application using *f.wmm* mode.
Close Browser	Minimizes the Netscape browser to an icon, regardless of which window has the focus. Uses *f.wmm* mode with embedded mouse movements.
Close Mail	Closes the Mailtool window but does not save changes.
Command Tool	Manages Commandtool application using *f.wmm* mode.
Compose	Presses the "Compose" button in Mailtool to begin writing new mail.
Day View	Uses captured mouse movement commands with *f.wmm* mode to change Calendar tool to day view.
Delete Message	Deletes the current message in the Mailtool application. Uses embedded

[8] Recall that *f.wmm* is the windows management mode as detailed in Chapter IV, p. 35.

{*Front:t:/usr/shell/mail/username*} command.

Deliver Presses the "Deliver" button in the Mailtool application composition window.

Done Presses the "Done" button in the Mailtool window. Saves changes made to the in-box

Exit Exits Frame Maker application.

File Manager Manages File Manager application using *f.wmm* mode.

Frame Maker Uses embedded mouse movements and the "*maker &*" command to give focus to the Commandtool window and to start Frame Maker.

Front Sends the current window to the front of the screen (or to the back of the screen if the window is currently at the front of the screen).

Help Starts the Help dialogue for OpenWindows 3. Uses *f.wmm* mode.

Icon Minimizes the current window to an icon.

Icon Editor Manages Icon Editor application using *f.wmm* mode.

In3 Manages IN^3 Voice Command application using *f.wmm* mode. Does not start IN^3. IN^3 must already be active.

Info Presses the "Info" button in Frame Maker.

Load In-Box Loads the Mailtool in-box.

Lower Sends the current window to a lower screen level.

Mail Manages Mailtool application using *f.wmm* mode.

Meter Manages CPU Meter tool application using *f.wmm* mode.

Month View Uses captured mouse movement commands with *f.wmm* mode to change Calendar tool to current month view.

Netscape Manages Netscape World Wide Web Browser application using *f.wmm* mode.

New CommandTool Uses *f.exec*[9] mode to start a new Commandtool window.

New Shell Tool Uses *f.exec* mode to start a new Shelltool window.

Next Message Views the next message in the Mailtool application. Uses embedded {*Front:t:/usr/shell/mail/username*} command.

[9] Recall that *f.exec* mode is the execution mode as detailed in Chapter IV, p. 36.

Ok Presses the “Ok” button when leaving Frame Maker

Open Presses the “Open” button in Frame Maker to open a document.

Open Location Presses the “File|Open location” menu selection in Netscape using embedded mouse movement commands.

Previous Message Views the previous message in the Mailtool application. Uses embedded {*Front:t:/usr/shell/mail/username*} command.

Print Presses the “Print” button in Netscape and the “Ok” button to print the current document.

Print Message Presses the “Print” dialogue menu selection in Mailtool using embedded mouse movement commands.

Print Tool Manages Printtool application using *f.wmm* mode.

Printer 14 Changes printer selection using *f.wmm* mode and embedded mouse movement commands.

Printer 2 Changes printer selection using *f.wmm* mode and embedded mouse movement commands.

Quit Audio Dismisses the Audiotool application window.

Quit Binder Dismisses the Binder application window.

Quit Browser Dismisses the Netscape browser application window.

Quit Command Tool Dismisses the Commandtool application window.

Quit Editor Dismisses the Text Editor application window.

Quit Icon Dismisses the Icon Editor application window.

Quit Mail Dismisses the Mailtool application window.

Quit Meter Dismisses the CPU Meter application window.

Quit Snapshot Dismisses the Snapshot application window.

Quit Tapetool Dismisses the Tapetool application window.

Quit Tool Dismisses the Shelltool application window.

Refresh Refreshes the workspace desktop in OpenWindows 3. Uses embedded mouse movement commands.

Reply Presses the “Reply” button in Mailtool to start the composition dialogue for replying to the currently selected message.

Save Workspace Uses captured mouse movement commands to save the current OpenWindows 3 workspace configuration.

scroll down Uses captured mouse movement commands to scroll the mailtool window down one page.

scroll up Uses captured mouse movement commands to scroll the mailtool window up one page.

Shell Tool Manages Shelltool application using *f.wmm* mode.

Signature Adds a signature to composed mail in Mailtool by using captured mouse movement commands to select the correct menu item.

Snapshot Manages Snapshot application using *f.wmm* mode.

Tape Tool Manages Tapetool application using *f.wmm* mode.

Text Editor Manages Text Editor application using *f.wmm* mode.

Today Uses captured mouse movement commands with *f.wmm* mode to change Calendar tool to the current day view.

View Message Views the current message in the Mailtool application. Uses embedded {*Front:t:/usr/shell/mail/username*} command.

Web Search Presses the "Net Search" button in the Netscape browser.

Week View Uses captured mouse movement commands with *f.wmm* mode to change Calendar tool to a current week view.

Year View Uses captured mouse movement commands with *f.wmm* mode to change Calendar tool to the current year view.

APPENDIX C. USERS GUIDE FOR DRAGONDICTATE FOR WINDOWS 1.3

This section will cover installation and setup tips as well as other useful suggestions not found or covered in the User's Guide or Quick Start Manual. Some of the suggestions are found in the manuals, but they are not very well documented.

A. INSTALLATION

When installing DragonDictate, the user is given the opportunity to either install everything or to install just the required files for a single user. I have found that it is better to install everything. Installing everything makes it a lot easier to add more users for the application. If you do not install everything, the software will request that you insert disk number five of the installation diskette set. Unless you happen to have this particular diskette handy (which I did not), then you must find the disk and insert it into the requested drive. This can be avoided by choosing to initially install everything. Doing so will make adding new users a snap. Whenever you wish to add a new user, the program will accomplish the necessary steps and the dialogue for creating a new user will begin after about 20 seconds.

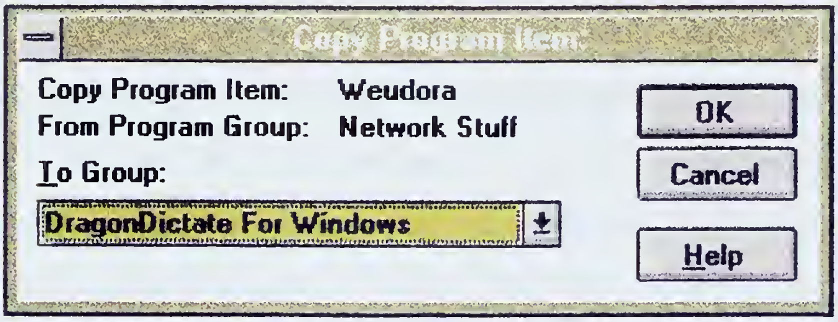
B. TRAINING

During the initial training, DragonDictates' training level is set at the default level which is light. This level enables the user to complete training with minimal time expended, but it offers the least amount of initial accuracy. This level only requires that the user repeat the word three times, whether the word recognized or not. It is recommended to set the level of training at intense initially. The intense level requires more repetitions of each word to be uttered by the user, but it offers the highest level of initial accuracy. The user is prompted to utter the command six times, and three more times if there is an error in the recognition of the word during the initial six utterances. This level setting takes a bit longer (about 45 minutes total time will be spent training),

but the improved accuracy and time not spent correcting errors is worth the extra time. Improvements in voice recognition speaker-independent models used in DragonDictate version 2.0 make initial training optional. The recognition of words is performed immediately after installation.

C. ADDING NON-SUPPORTED WINDOWS APPLICATIONS

Adding non-supported applications to the can be accomplished in three ways: 1) dragging the application icon to the DragonDictate For Windows group and dropping it, 2) adding the group using the Program Manager "File|New|Program Item" menu selection while the DragonDictate for Windows group is open, or 3) by copying a program item from one group to the DragonDictate For Windows group using the Program Manager "Copy Program item" menu selection and dialogue. (Figure 20). In either case you will have to possibly rename the program icon using the Program Manager "File|Properties" menu selection. As shown in Figure 20, It would be preferable to rename "Weudora" to "Eudora."


Copy Program Item

Copy Program Item: Weudora
From Program Group: Network Stuff
To Group:
DragonDictate For Windows

Figure 21. Copy Program Item dialogue window[10]

After copying the application it will be necessary to train the non-supported application command. This is accomplished by opening DragonDictates vocabulary manager and choosing the "Find Word" button. The word in this case will be "[Eudora]". All commands in DragonDictate are enclosed in brackets. Once the

[10] This dialogue is opened by choosing the "File|Copy" menu selection from the Windows Program Manager main window menu.

command is located, then click on the train command button. This will begin the training process for the non-supported application command. Once this is completed you will be able to use the "[Bring Up]" command to start the non-supported application by voice. All of the vocabulary for the menu items in the non-supported application can also be accessed by voice. DragonDictate will be able to track these automatically.

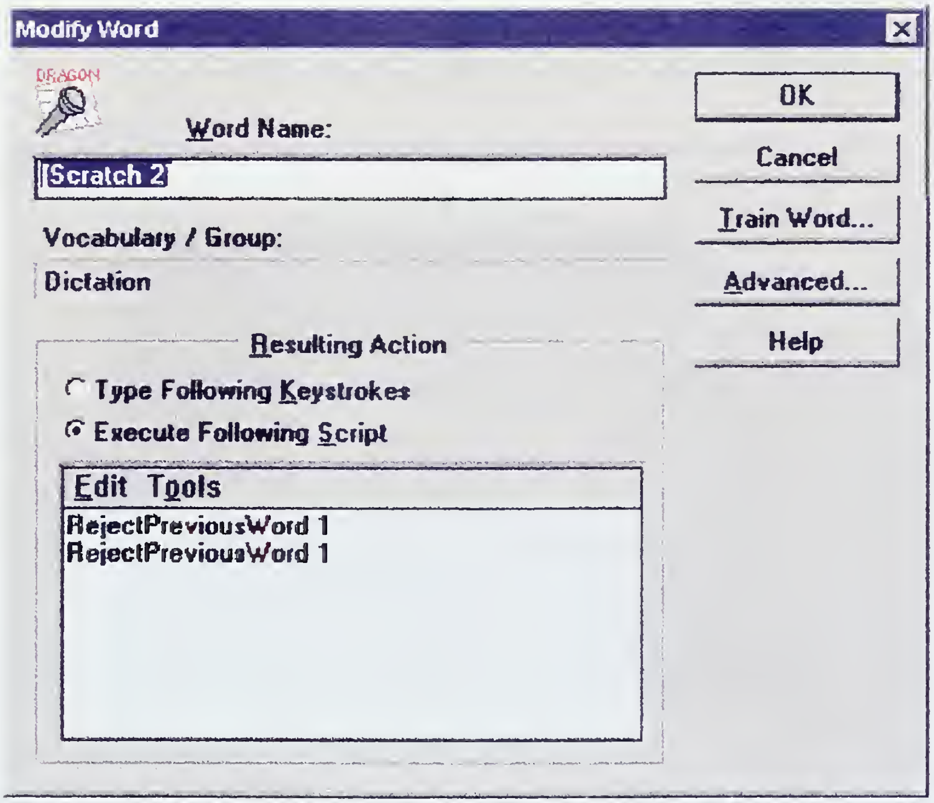
D. ADDING VOCABULARY FOR UPGRADED APPLICATIONS

Adding vocabulary for upgraded applications is very simple, though the User's Guide does not address this problem. For our example let us use an upgrade from WordPerfect 6.0 (there is an existing default vocabulary installed with the program) to Wordperfect 6.1 (there is no supporting vocabulary for this application). When Wordperfect 6.1 is opened using DragonDictate a vocabulary called WPWin 6.1 is added to the list of vocabularies in the vocabulary manger. Simply use the vocabulary manager to export this vocabulary as a text file. This is done by using the import/export vocabulary method described in the User's Guide. Simply name the file "WP61.txt". Export the WordPerfect 6.0 vocabulary and call it "WP60.txt." Open "WP60.txt" using Notepad, or any other text editor, and copy the entire document with the exception of the first two lines. Next, open "WP61.txt" and paste the text copied from the first document into this document after the two lines that are already present in "WP61.txt". Close both documents. Use vocabulary manager to import "WP61.txt" back into DragonDictate. Now all of the voice commands from WordPerfect 6.0 are available to WordPerfect 6.1.

E. CREATING NEW COMMANDS

Creating new commands is very simple in DragonDictate. Using the method described in the User's Guide, it is possible to develop custom commands. The Commands "[Scratch 2]" and others were created by modifying the command "[Scratch That]". By copying and pasting the resulting action from "[Scratch That]" it was possible to create commands to delete multiple words. By changing the "Resulting Action" text to

include the line “RejectPreviousWord 1” (Figure 21) multiple times or by adding a 2, 3, 4, or 5 instead of a 1, it is possible to create commands to delete multiple words.



Word Name: [Scratch 2]

Vocabulary / Group: Dictation

Resulting Action

○ Type Following Keystrokes

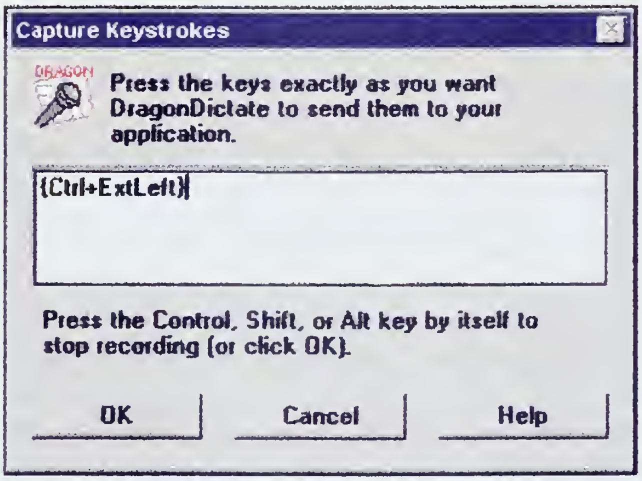
● Execute Following Script

```
RejectPreviousWord 1
RejectPreviousWord 1
```

Figure 22. Adding a new command “[Scratch 2].”

By inserting keystrokes instead of scripts it is possible to add other commands. The commands “[Back]”, “[Forward]”, and “[Reload]” were created for Netscape Navigator using this method (Figure 22). The dialogue window is able to capture the required keystrokes by choosing the “Tools|Capture Keystrokes” menu selection in the “Resulting Action” box captures the keystrokes that are performed by the user. The keystrokes are then transferred to the resulting action box (Figure 23). Using these two

methods allows the user to add custom commands to augment the default commands for applications, and to add to the command vocabulary for non-supported applications. This allows DragonDictates 30,000 word vocabulary to be tailored to fit the requirements of the user. The vocabulary does not expand or increase. Words that are not used are simply dropped out of the vocabulary to make room for the new words.



Capture Keystrokes

Press the keys exactly as you want DragonDictate to send them to your application.

{Ctrl+ExtLeft}

Press the Control, Shift, or Alt key by itself to stop recording (or click OK).

Figure 23. Capturing Keystrokes

APPENDIX D.   DICTATION TEST PARAGRAPH

The following passage from "Of the Standard of Taste" by David Hume was used as a control to measure the accuracy and learning capacity of DragonDictate [Ref. 17: p. 210]:

> The great resemblance between mental and bodily taste will easily teach us to apply this story. Though it be certain, that beauty and deformity, more than sweet and bitter, are not qualities in objects, but belong entirely to the sentiment, internal or external; it must be allowed, that there are certain qualities in objects, which are fitted by nature to produce those particular feelings. Now as these qualities may be found in a small degree, or may be mixed and confounded with each other, it often happens that the taste is not affected with such minute qualities, or is not able to distinguish all the particular flavours, amidst the disorder in which they are presented. Where the organs are so fine, as to allow nothing to escape them; and at the same time so exact, as to perceive every ingredient in the composition: This we call delicacy of taste, whether we employ these terms in the literal or metaphorical sense. Here then the general rules of beauty are of use, being drawn from established models, and from the observation of what pleases or displeases, when presented singly and in a high degree: And if the same qualities, in a continued composition, and in a smaller degree, affect not the organs with a sensible delight or uneasiness, we exclude the person from all pretensions to this delicacy. To produce these general rules or avowed patterns of composition, is like finding the key with the leathern thong; which justified the verdict of Sancho's kinsmen, and confounded those pretended judges who had condemned them. Though the hogshead had never been emptied, the taste of the one was still equally delicate, and that of the other equally dull and languid: But it would have been more difficult to have proved the superiority of the former, to the conviction of every bye-stander....

LIST OF REFERENCES

1. Dragon Dictate for Windows 1.0 User's Guide, first edition, Dragon Systems, Inc., Newton, Massachusetts, 1994.

2. Dragon Dictate for Windows 1.0 Getting Started, first edition, Dragon Systems, Inc., Newton, Massachusetts, 1994.

3. Microsoft® Windows Sound System 2.0 Software User's Guide, Microsoft® Corporation, Redmond, Washington, 1993.

4. Rash, Ray Jr., "Talk Show: Voice-recognition technology for the PC lets your voice run the show," PC Magazine, December 20, 1994.

5. Rudnicky, Alexander I., Hauptmann, Alexander G., and Lee, Kai-Fu. "Survey of current speech technology," Communications on the ACM, March 1994, Vol. 37 No. 3.

6. Remich, Norman C. Jr., "Talking to Appliances." Appliance Manufacturer, July 1993. Vol. 41 No. 7.

7. "Military, civil applications seen in USAF voice identification system." Aviation Week & Space Technology, April 6, 1992 vol. 136 No. 14.

8. "New Speech-Recognition Systems Turn Up The Application Volume." InSide Gartner Group This Week. Nov. 9, 1994 vol. X, No. 45.

9. "Speech recognition: BBN intros 2nd generation of HARK products. " EDGE, On & About AT&T Sept. 26, 1994 Vol. 9.

10. Hill, Earl T., and Kotowski, Leo B. Using Voice Recognition As An Input Medium To The JINTAACS Automated Message Preparation System (JAMPS), Masters Thesis, Naval Postgraduate School, Monterey, California March, 1986.

11. Stanco, B. "Voice and Call Processing." Inside Gartner Group This Week, November 4, 1994 vol. X, No. 44.

12. Keyes, Jessica. The McGraw-Hill Multimedia Handbook, McGraw-Hill, Inc., New York, New York, 1994.

13. World Wide Web site- *http://www-speech.sri.com/projects/hybrid.html*, dated 6/12/95.

14. O'Shaughnessy, D. Speech Communication, Addison Wesley, 1987.

15. IN^3 Voice Command User's Guide For SPARCstation, Command Corp., Duluth, Georgia, 1994.

16. World Wide Web site - *http://www.cse.ogi.edu/SLAM/*, dated 1/19/96.

17. "English Essays: From Sir Philip Sydney to McCauley." The Harvard Classics. Grolier Enterprises Corp., Danbury, Connecticut, 1980.

18. Director of Command, Control, Communications, and Computer Systems, J-6, Committed, Focused, and Needed, C4I For The Warrior, The Joint Staff, Department of Defense, Washington, D.C., June 12, 1993.

INITIAL DISTRIBUTION LIST

	No. Copies
1. Defense Technical Information Center 8725 Kingman Rd., STE 0944 Ft. Belvoir, VA 22060-6218	2
2. Dudley Knox Library Naval Postgraduate School 411 Dyer Rd. Monterey, California 93943-5101	2
3. Prof. Monique P. Fargues, Code EC/FA Naval Postgraduate School Monterey, California 93943-5101	2
4. Chairman, Code EC ECE Department Naval Postgraduate School Monterey, CA 93943-5101	1
5. Prof. James C. Emery, Code SM/EY Naval Postgraduate School Monterey, California 93943-5101	2
6. Prof. Roger Stemp, Code CS/SP Naval Postgraduate School Monterey, California 93943-5101	1
7. LT Timothy J. West Old Wagner Rd Petersburg, VA 23805	2
8. LT Hugh P. Brien 8661 Point of Woods Dr. Manassas, VA 22110	1